



MASTERARBEIT

Herr
Markus Dod

**Zuverlässigkeit in gerichteten
Netzwerken**

2012

MASTERARBEIT

Zuverlässigkeit in gerichteten Netzwerken

Autor:

Markus Dod

Studiengang:

Diskrete- und Computerorientierte Mathematik

Seminargruppe:

ZD10

Erstprüfer:

Prof. P. Tittmann

Zweitprüfer:

Prof. S. Helbig

Mittweida, 2012

Bibliografische Angaben

Dod, Markus: Zuverlässigkeit in gerichteten Netzwerken, 81 Seiten, 41 Abbildungen, Hochschule Mittweida (FH), Fakultät MNI

Masterarbeit, 2012

Dieses Werk ist urheberrechtlich geschützt.

Referat

Diese Arbeit beschäftigt sich mit verschiedenen Zuverlässigkeitsproblemen in gerichteten Netzwerken. Dabei wird speziell die s,t -Zuverlässigkeit und die s,T -Zuverlässigkeit betrachtet. Dazu werden verschiedene Berechnungs- und Reduktionsmöglichkeiten vorgestellt und anhand von Testrechnungen miteinander verglichen. Außerdem werden für spezielle Graphenklassen explizite und rekursive Formeln angegeben.

Danksagung

Ein herzliches Dankeschön geht an alle, die mich bei der Erstellung meiner Masterarbeit unterstützt haben. Besonders möchte ich mich bei Prof. Peter Tittmann bedanken, der mich fortwährend unterstützt hat.

Ganz besonders bedanken möchte ich mich bei meiner Frau Ester, denn ohne deine moralische Unterstützung wäre ich niemals fertig geworden. Vielen Dank auch für das Korrekturlesen und die Hilfe bei schwierigen Formulierungen.

I. Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	II
Tabellenverzeichnis	III
1 Einleitung	1
2 Grundlagen und Vorbetrachtungen	3
2.1 Graphentheoretische Grundlagen	3
2.2 Arten von Zuverlässigkeitsproblemen	4
2.3 Komplexität von Zuverlässigkeitsproblemen	5
2.4 Literatur	8
3 s,t-Zuverlässigkeit	11
3.1 Reduktionen	11
3.2 Artikulationen und Brücken	17
3.3 Exakte Algorithmen	18
3.3.1 Vollständige Enumeration	19
3.3.2 Inklusion-Exklusion	20
3.3.3 Dekomposition	21
3.3.4 Berechnung mit Hilfe von Schnitten	23
3.4 Zuverlässigkeit in speziellen Graphenklassen	31
3.4.1 Vollständige Graphen	31
3.4.2 Vollständige bipartite Graphen	33
3.4.3 Fächergraphen	34
3.4.4 Radgraphen	37
3.4.5 Gittergraphen	39
3.4.6 Fibonaccigraphen	40
3.4.7 BSP-Graphen	42
3.4.8 Graphen mit beschränkter Wegweite	44

3.4.9	Azyklische Graphen	51
4	s,T-Zuverlässigkeit	55
4.1	Reduktionen	55
4.2	Berechnungsverfahren	56
4.2.1	Vollständige Enumeration	56
4.2.2	Inklusion-Exklusion	57
4.2.3	Dekomposition	58
4.2.4	Graphen mit beschränkter Wegweite	59
4.2.5	Graphen mit beschränkter Baumweite	61
5	Implementation und Testrechnungen	65
5.1	Implementation	65
5.1.1	Ablauf einer Berechnung	65
5.2	Testrechnungen	66
6	Auswertung, Schlussfolgerungen und offene Probleme	69
A	Beispielgraphen	71
	Literaturverzeichnis	75

II. Abbildungsverzeichnis

2.1	Knoten mit Ausfallwahrscheinlichkeit durch Bogen ersetzen	4
2.2	Transformation für s,t-Zuverlässigkeit	6
3.1	Parallel-, Serien- und Serien-Parallel-Reduktion	11
3.2	Arten von Knoten mit zwei Nachbarn	12
3.3	Kanten-Reduktion	13
3.4	Reduktion durch Entfernen unnötiger Knoten und Bögen	13
3.5	s-Dreiecks-Reduktion	14
3.6	Dekompositionsbaum für den Graphen G	15
3.7	Dekompositionsbaum für den Graph G'	16
3.8	t-Dreiecks-Reduktionen	16
3.9	Artikulation	17
3.10	Brücke	18
3.11	Mehrfaches passieren der trennenden Knotenmenge	19
3.12	Graph mit 4 Knoten	20
3.13	Problem bei der Kontraktion eines Bogens	22
3.14	Dekompositionsbaum	23
3.15	Beispiel für die Mengen in Schnittelalgorithmus	25
3.16	Vollständiger Graph mit fünf Knoten	31
3.17	Vollständiger bipartiter Graph mit sechs Knoten	33
3.18	Fächergraph	35
3.19	Fächergraph	36
3.20	Radgraph mit acht Knoten	37
3.21	Gittergraph	39
3.22	Gittergraph	40
3.23	Fibonacci-Graph	40
3.24	Graph mit der Wegweite zwei	46
3.25	Trennende Knotenmenge in einem Graphen	52
3.26	Beispielgraph	53

4.1	Konstellation bei der keine Serienreduktion möglich ist	55
4.2	Reduktionen bei der Berechnung der s,T-Zuverlässigkeit	55
4.3	Reduktionen bei der Berechnung der s,T-Zuverlässigkeit	56
4.4	Graph mit zwei Nichtterminalknoten und zwei Terminalknoten	57
4.5	Dekompositionsbaum	59
4.6	Baumzerlegung eines Graphen	62
4.7	Verknüpfung zweier Graphen	63
A.1	FM-Graph	71
A.2	Satyanarayana-Graph	71
A.3	Petersen-Graph	72
A.4	Prosemgraph	72
A.5	Satyanarayana-Prabhakar-Graph	73
A.6	PC-Graph	74

III. Tabellenverzeichnis

2.1	Komplexität von Zuverlässigkeitsproblemen [Col87]	8
3.1	Berechnung der Schnitte im Standardgraphen	30
3.2	Zuverlässigkeitspolynome von vollständigen Graphen	32
3.3	Zuverlässigkeitspolynome von vollständig bipartiten Graphen	35
3.4	Ablauf des Graphenzerlegungsalgorithmus	49
3.5	Anzahl der azyklischen Graphen mit n Knoten	49
4.1	Ablauf des Graphenzerlegungsalgorithmus zur Berechnung der s,T -Zuverlässigkeit . .	61
4.2	Ablauf des Baumzerlegungsalgorithmus zur Berechnung der s,T -Zuverlässigkeit . . .	64
5.1	Liste der verwendeten Testgraphen	66
5.2	Laufzeiten der Testverfahren in s	67

1 Einleitung

Durch den vermehrten Einsatz von netzbasierten Anwendungen spielen die Zuverlässigkeit von Computer- und Kommunikationsnetzwerken eine immer entscheidendere Rolle für Unternehmen und auch Privatpersonen. Durch die zunehmende Integration von IT-Systemen entlang der Wertschöpfungskette wächst die Abhängigkeit von zuverlässigen und störungsfrei operierenden Kommunikationsnetzwerken. Ein typisches Beispiel sind Computernetzwerke. Hier muss die Kommunikation zwischen einzelnen Computern oder Servern sichergestellt sein. In der Realität können allerdings Verbindungen und einzelne Komponenten im Computernetz mit einer gewissen Wahrscheinlichkeit ausfallen. Deshalb müssen solche Netze so geplant werden, dass die Wahrscheinlichkeit für den Ausfall des Systems oder bestimmter Funktionsmerkmale gering ist. Diese Funktionsfähigkeit ist besonders bei neuen Leistungsmerkmalen, wie dem Voice over IP (VoIP) oder Video- und Fernsehdiensten über Netzwerke wichtig. Auch die Entwicklung im Bereich des *Intelligenten Hauses* (die Geräte im Haus sind miteinander vernetzt, können miteinander kommunizieren und von außen ferngesteuert werden) stellt neue Anforderungen an die Zuverlässigkeit von Kommunikationsnetzen. Diese Dienste müssen ständig und ausfallsicher verfügbar sein. Um konkrete Aussagen über die Zuverlässigkeit eines Netzwerkes treffen zu können, wird dieses mit Hilfe von Graphen modelliert. Auf ihnen werden dann die entsprechenden Kenngrößen berechnet. Besonders interessiert hier die Wahrscheinlichkeit, dass jeder Knoten mit jedem anderen Knoten des Netzwerkes kommunizieren kann und die Wahrscheinlichkeit, dass ein bestimmter Knoten mit einem anderen Knoten des Netzwerkes in Verbindung treten kann. Daneben sind noch eine Vielzahl weiterer Zuverlässigkeitsprobleme modellierbar, die besondere Anforderungen an ein funktionierendes System berücksichtigen. In der Vergangenheit wurden vor allem für ungerichtete Netzwerke tiefgehende Untersuchungen über die Berechnung von Zuverlässigkeitsmaßen angestellt. In dieser Arbeit wird die Berechnung in gerichteten Netzwerken näher untersucht und Verfahren dazu beschrieben. Ein Schwerpunkt dieser Arbeit liegt dabei auf der Angabe von expliziten Lösungsformeln für spezielle Graphenklassen und die effiziente Lösung in Graphen mit geringer Wegbeziehungsweise Baumweite.

Konkret wird in dieser Arbeit die s, t - und die s, T -Zuverlässigkeit, also die Wahrscheinlichkeit, dass ein intakter gerichteter Weg vom Knoten s zum Knoten t , beziehungsweise zu allen Knoten aus der Menge T existiert, untersucht. Um eine einheitliche Basis zu schaffen, vermittelt Kapitel 2 die graphentheoretischen Grundlagen und bietet eine Übersicht über die verschiedenen Zuverlässigkeitsprobleme und ihre Komplexität. Anschließend wird in Kapitel 3 die Berechnung der s, t -Zuverlässigkeit beschrieben. An dieser Stelle erwarten den Leser verschiedene Möglichkeiten der Reduktion von Graphen und somit eine Möglichkeit, die Berechnung der Zuverlässigkeit zu beschleunigen. Außerdem werden verschiedene Graphenklassen vorgestellt, in denen die Berech-

nung der s,t-Zuverlässigkeit einfach möglich ist. Kapitel 4 bietet einen Überblick über das s,T-Zuverlässigkeitsproblem. Die vorgestellten Algorithmen zur Berechnung der s,t-Zuverlässigkeit wurden in Python implementiert. Eine Beschreibung der Implementation und die Ergebnisse der Testrechnungen der verschiedenen Verfahren sind in Kapitel 5 zu finden. Das letzte Kapitel fasst die Ergebnisse dieser Arbeit noch einmal zusammen und bietet zudem einen Ausblick auf die offenen Probleme der Zuverlässigkeit in Netzwerken.

2 Grundlagen und Vorbetrachtungen

In diesem Kapitel werden sowohl einige graphentheoretische Grundlagen beschrieben, als auch eine Übersicht über die verschiedenen Zuverlässigkeitsprobleme gegeben. Im Abschnitt 2.3 werden Untersuchungen über die Komplexität von verschiedenen Zuverlässigkeitsproblemen vorgenommen. Der letzte Abschnitt dieses Kapitels bietet einen Einblick in die vorhandene Literatur zu diesem Themengebiet.

2.1 Graphentheoretische Grundlagen

In diesem Abschnitt soll kurz auf die graphentheoretischen Grundlagen eingegangen werden, die dieser Arbeit zugrunde liegen. Eine ausführliche Einführung in die Grundbegriffe der Graphentheorie sind in [Gru95] und [BJ08] zu finden.

Definition 2.1 Ein ungerichteter Graph $G = (V, E)$ besteht aus einer Knotenmenge V und einer Kantenmenge $E \subseteq V \times V$, wobei jeder Kante $e \in E$ von G zwei (nicht notwendig verschiedene) Knoten aus V zugeordnet sind. Ein gerichteter Graph oder auch Digraph $G = (V, E)$ besteht aus einer Knotenmenge V und einer Bogenmenge $E \subseteq V \times V - \{(x, x) | x \in V\}$ von geordneten Knotenpaaren.

Definition 2.2 Zwei Bögen $e = (u_1, v_1)$ und $f = (u_2, v_2)$ in einem Graphen heißen parallel, falls $u_1 = v_1$ und $u_2 = v_2$ gilt und sie heißen antiparallel, falls $u_1 = v_2$ und $u_2 = v_1$ gilt. Ein Bogen $e = (v, v)$ heißt Schlinge. Ein Graph ohne Schlingen und parallele Bögen heißt schlichter Graph. Im Folgenden werden nur schlichte Graphen betrachtet.

Definition 2.3 Ein gerichteter Weg in einem Graphen $G = (V, E)$ ist eine Folge von Knoten $(v_1, v_2, v_3, \dots, v_k)$ des Graphen, mit $v_i \neq v_j$ für $i \neq j$ und $(v_i, v_{i+1}) \in E$ für $i = 1, \dots, k-1$. Ein gerichteter Kreis ist ein Weg, für den gilt: $v_1 = v_k$ und $v_i \neq v_j$ für $1 < i, j < k$.

Der Innengrad $d^-(v)$ des Knotens v ist die Anzahl der in den Knoten v eingehenden Bögen, und der Außengrad $d^+(v)$ des Knotens ist die Anzahl der ausgehenden Bögen. Ein Graph heißt azyklisch, falls er keine gerichteten Kreise besitzt.

Definition 2.4 Ein gerichtetes, stochastisches Netzwerk ist ein gerichteter Graph, bei dem jeder Bogen e , beziehungsweise jeder Knoten v , mit einer Wahrscheinlichkeit $0 \leq p_e \leq 1$ versehen ist. Die Bögen und Knoten fallen unabhängig voneinander aus.

In dieser Arbeit werden ausschließlich gerichtete Netzwerke mit Bögenausfällen betrachtet. Die Knoten werden als ausfallfrei angesehen. Sind die Knoten in einem Netzwerk auch ausfallbehaftet, kann dieser Graph in einen Graphen ohne Knotenausfälle

überführt werden. Das Prinzip dieser Umwandlung ist in der Abbildung 2.1 zu sehen. Der Knoten u ist mit der Wahrscheinlichkeit p_u verfügbar. Der Knoten u wird durch zwei neue Knoten v und w ersetzt. Alle Bögen, die in den Knoten u hineingeführt haben, gehen jetzt in den Knoten v hinein und alle Bögen, die aus dem Knoten u herausgingen, gehen jetzt aus dem Knoten w heraus. Die beiden Knoten v und w werden anschließend noch mit einem Bogen verbunden, der mit dem Gewicht des Knotens u , p_u , gewichtet wird. Für jeden Knoten des Graphen wird somit ein zusätzlicher Knoten und ein zusätzlicher Bogen eingefügt. Ein Graph mit n Knoten und m Bögen hat nach der Ersetzung der ausfallbehafteten Knoten somit $2n$ Knoten und $m + n$ Bögen.

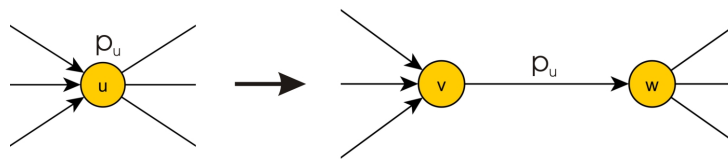


Abbildung 2.1: Knoten mit Ausfallwahrscheinlichkeit durch Bogen ersetzen

2.2 Arten von Zuverlässigkeitsproblemen

In stochastischen Netzwerken sind einige Kenngrößen von hoher praktischer Relevanz. So interessiert besonders die Wahrscheinlichkeit, dass zwei Knoten in einem solchen Netzwerk miteinander kommunizieren können, also mindestens eine intakte Verbindung zwischen den beiden Knoten existiert. Dieses Problem wird auch als s,t-Zuverlässigkeit bezeichnet. Im Englischen ist es in ungerichteten Graphen unter dem Namen „s,t-reliability“ in der Literatur zu finden. Im Folgenden wird für dieses Problem in ungerichteten Graphen die Abkürzung Rel_2 und in gerichteten Graphen die Abkürzung $Conn_2$ (engl. s,t-connectness) verwendet. Im gerichteten Fall ist die Wahrscheinlichkeit gesucht, dass ein intakter, gerichteter s,t-Weg existiert.

Die s,t-Zuverlässigkeit kann auf mehrere Knoten verallgemeinert werden. Im ungerichteten Fall wird dieses Problem k-Terminal-Zuverlässigkeit (engl. k-terminal reliability) $Rel_k(G)$ genannt. Hier wird die Wahrscheinlichkeit gesucht, dass alle k Knoten dieser Mengen miteinander verbunden sind. Es gibt zwei Möglichkeiten, dieses Problem auf den gerichteten Fall zu verallgemeinern. Beim ersten Fall wird das Problem s,T-Zuverlässigkeit (engl. s,T-connectness) $Conn_{sT}(G)$ genannt. Hierbei ist die Wahrscheinlichkeit gesucht, dass vom Knoten s aus gerichtete Wege zu allen $k - 1$ Knoten der Menge T existieren. Die zweite Möglichkeit ist die K-Terminal-Zuverlässigkeit (engl. K-terminal reliability) $Conn_K(G)$. Hier ist die Wahrscheinlichkeit gesucht, dass alle Terminalknoten der Terminalknotenmenge K miteinander kommunizieren können. Es müssen also intakte, gerichtete Wege von allen Knoten aus K zu allen anderen Knoten der Menge K existieren.

In manchen Fällen wird aber auch die Wahrscheinlichkeit gesucht, dass jeder Knoten

des Graphen mit jedem anderen Knoten über einen Weg verbunden ist. Im ungerichteten Fall bedeutet dies, dass der Graph zusammenhängend ist. Dieses Problem wird Zusammenhangswahrscheinlichkeit (engl. all-terminal reliability) $Rel_A(G)$ genannt. Im gerichteten Fall ist die Wahrscheinlichkeit gesucht, dass der Graph streng zusammenhängend ist, das heißt, dass von jedem Knoten aus ein gerichteter Weg zu jedem anderen Knoten des Graphen existiert. Dieses Problem wird Erreichbarkeit (engl. reachability oder auch strongly connected network problem) $Conn_A(G)$ genannt.

Neben diesen sieben hier aufgeführten Zuverlässigkeitsproblemen können noch weitere definiert werden. Ein allgemeingültiges Verfahren, um weitere Zuverlässigkeitsprobleme zu definieren, benutzt die Enumeration von Zuständen. Ein Zustand S ist eine Teilmenge der Bogenmenge E . Dabei werden die Bögen der Menge S als die intakten interpretiert und die der Menge $E - S$ als die ausgefallenen. Es gibt somit $2^{|E|}$ verschiedene Zustände. Eine Netzwerkoperation $OP(G)$ ist eine Menge mit $OP(G) \subseteq 2^E$. In der Menge $OP(G)$ sind die Zustände enthalten, bei denen das System funktionsfähig ist. Die Menge $OP(G)$ wird auch stochastisches, binäres System (SBS) genannt. Die Elemente der Menge heißen dabei Wegmenge. Mit Hilfe dieser Wegmengen kann das Zuverlässigkeitspolynom berechnet werden. Eine Wegmenge mit i Knoten tritt dabei mit der Wahrscheinlichkeit $p^i(1-p)^{m-i}$, mit $|E| = m$, auf. N_i sei die Anzahl der i -elementigen Wegmengen, dann wird das folgende Polynom das Zuverlässigkeitspolynom genannt.

$$Rel(G, p) = \sum_{i=0}^m N_i p^i (1-p)^{m-i}$$

Um eine bestimmte Wahrscheinlichkeit zu ermitteln, muss für p die entsprechende Verfügbarkeitswahrscheinlichkeit der Bögen eingesetzt werden. Mit diesem Verfahren können auch sehr spezielle Zuverlässigkeitsprobleme modelliert werden.

2.3 Komplexität von Zuverlässigkeitsproblemen

Um Aussagen über die Komplexität von Zuverlässigkeitsproblemen treffen zu können, muss die Komplexitätsklasse $\#P$ eingeführt werden. Diese Klasse wurde von L. G. Valiant in [Val79] erstmalig definiert. Die folgende Definition stammt von I. Wegener [Weg03].

Definition 2.5 Die Komplexitätsklasse $\#P$ enthält alle Anzahlprobleme A , für die es eine polynomiell zeitbeschränkte, nichtdeterministische Turingmaschine gibt, die für jede Eingabe x so viele akzeptierende Rechenwege hat, wie es Lösungen für x gibt.

Analog zu den Klassen NP und NP-vollständig, kann auch für die Klasse $\#P$ die Klasse $\#P$ -vollständig eingeführt werden.

Definition 2.6 Ein Problem A ist $\#P$ -vollständig, genau dann wenn

$$A \in \#P \wedge B \in \#P : B \leq_p A$$

gilt.

Mit Hilfe dieser Definition kann nun eine Aussage über die Komplexität des s,t -Zuverlässigkeitsproblems gemacht werden. Aussagen für die anderen Teilprobleme der Zuverlässigkeitstheorie in ungerichteten und gerichteten Graphen sind in [Col87] zu finden.

Satz 2.7 Das Problem *BIPARTITE INDEPENDENT SET* ist in polynomialer Zeit reduzierbar auf die Zählung der minimalen s,t -Schnittmengen eines ungerichteten Graphen.

Beweis: Eine Instanz von *BIPARTITE INDEPENDENT SET* besteht aus einem bipartiten Graphen $G = (V, E)$, mit $X \cap Y = \emptyset$. Nun wird ein neuer Graph $H = (W, F)$ aus G konstruiert. In der Knotenmenge W des Graphen befinden sich alle Knoten des Graphen G , zusätzlich sind zwei Knoten s und t enthalten. Die Kantenmenge F des Graphen H enthält alle Kanten in E . Zusätzlich werden zu jedem Knoten aus der Teilmenge X k parallele Kanten zum Knoten s eingefügt. Dabei ist k der Knotengrad des entsprechenden Knotens in X . Genauso werden zwischen jedem Knoten aus der Knotenteilmenge Y zum Knoten t k parallele Kanten eingefügt. Diese Transformation ist in Abbildung 2.2 zu sehen.

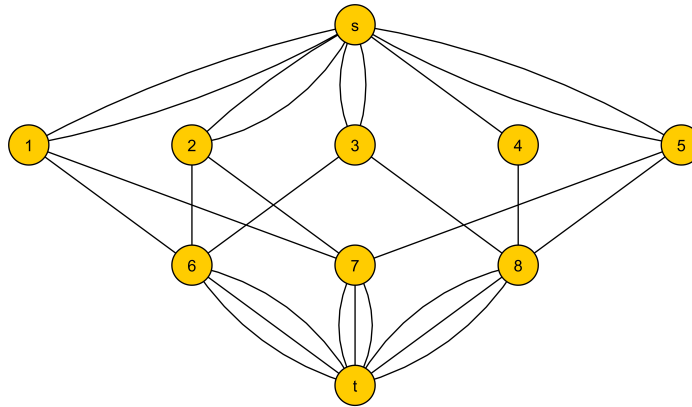


Abbildung 2.2: Transformation für s,t -Zuverlässigkeit

Die Konstruktion sichert, dass ein s,t -Schnitt mindestens $|E|$ Kanten enthält und dass es Schnitte gibt, die nur $|E|$ viele Kanten enthalten. Nun wird eine eindeutige Zuordnung zwischen den s,t -Schnitten mit $|E|$ Kanten in H und unabhängigen Mengen in G hergestellt. Als Erstes kann festgestellt werden, dass falls $k \geq 1$ Kanten zwischen einem Knoten aus X und s bestehen, entweder alle Kanten in einem Schnitt sind oder keine der k Kanten. Falls nur eine der k Kanten enthalten wäre, könnte diese Kante aus dem Schnitt entfernt werden und es würde dadurch ein kleinerer Schnitt entstehen. Weiterhin gilt, dass falls Kanten $\{s, x\}$ und $\{y, t\}$ in einem minimalen Schnitt enthalten sind, dann ist $\{x, y\}$ keine Kante in G . Wäre nämlich $\{x, y\}$ eine Kante in G , so könnte durch Entfer-

nen aller parallelen Kanten $\{s, x\}$ und $\{y, t\}$ vom Schnitt und Einfügen aller Kanten in G , die x oder y enthalten, ein Schnitt konstruiert werden, der eine Kante weniger enthält. Dies widerspricht der Annahme, dass der Schnitt minimal ist.

Alle Kanten in einer minimalen Schnittmenge von H , welche entweder s oder t enthalten, definieren eine Teilmenge der Knotenmenge von G , welche eine unabhängige Menge ist. Andererseits definiert jede unabhängige Knotenmenge in G einen minimalen s, t -Schnitt. \square

Mit Hilfe dieses Satzes kann jetzt eine Aussage über die Komplexität des s, t -Zuverlässigkeitsproblems Rel_2 in ungerichteten Graphen getroffen werden.

Folgerung 2.8 *Das s, t -Zuverlässigkeitsproblem Rel_2 in ungerichteten Graphen ist $\#P$ -vollständig.*

Beweis: Um die Anzahl der Wegmenge von s nach t zu berechnen, genügt es, die Anzahl der minimalen s, t -Schnitte zu bestimmen. \square

Um aus dem Satz 2.7 eine Aussage über die Komplexität des s, t -Zuverlässigkeitsproblems $Conn_2$ in gerichteten Graphen treffen zu können, wird ein Lemma über die Überführbarkeit von Rel_k und $Conn_k$ benötigt.

Lemma 2.9 $Rel_k \leq_p Conn_k$

Folgerung 2.10 *Das s, t -Zuverlässigkeitsproblem $Conn_2$ in gerichteten Graphen ist $\#P$ -vollständig.*

Beweis: Dies folgt aus dem Satz 2.7 und dem Lemma 2.9. \square

Manche der in Kapitel 3 angegebenen Verfahren beruhen auf der Enumeration von minimalen Wegen bzw. Schnitten im Graphen. Falls alle Minimalschnitte in einem Graphen gegeben sind, kann das s, t -Zuverlässigkeitsproblem in Abhängigkeit von der Schnittanzahl, in polynomialer Zeit gelöst werden (siehe Abschnitt 3.3.4). Für die minimalen s, t -Wege im Graphen gilt dies nicht.

Satz 2.11 *Das s, t -Zuverlässigkeitsproblem in gerichteten Graphen bleibt $\#P$ -vollständig, selbst wenn alle Minimalwege gegeben sind.*

Beweis: Für den Beweis wird der in Satz 2.7 konstruierte Graph verwendet. In diesem Graphen seien alle Kanten von s nach t gerichtet. Die Anzahl der Minimalwege in diesem Graphen ist polynomial in der Größe des gerichteten Graphen. Somit ist die Ermittlung aller Wege eine Instanz des Problems, die polynomial in der Größe des Originalgraphen ist. \square

In der folgenden Tabelle wird eine Übersicht über die Komplexität von verschiedenen Zuverlässigkeitsproblemen in ungerichteten und gerichteten Graphen gegeben. Die Komplexität ist für die Zuverlässigkeitsprobleme im Graphen und bei gegebenen Minimalwegen, bzw. Minimalschnitten angegeben.

Problem	Enumeration	Minimalwege	Minimalschnitte
<i>ungerichteter Graph</i>			
s,t-Zuverlässigkeit Rel_2	#P-vollständig	offen	polynomial
s,T-Zuverlässigkeit Rel_k	#P-vollständig	offen	offen
Zusammenhangsw. Rel_A	#NP-vollständig	polynomial	offen
<i>gerichteter Graph</i>			
s,t-Zuverlässigkeit $Conn_2$	#P-vollständig	#P-vollständig	polynomial
s,T-Zuverlässigkeit $Conn_k$	#P-vollständig	#P-vollständig	#P-vollständig
Erreichbarkeit $Conn_A$	#P-vollständig	polynomial	offen

Tabelle 2.1: Komplexität von Zuverlässigkeitsproblemen [Col87]

2.4 Literatur

Zuverlässigkeitsprobleme in stochastischen Netzwerken wurden von verschiedenen Autoren intensiv untersucht. Eine gute Einführung in die Zuverlässigkeitsberechnung in Graphen ist in „The Combinatorics of network reliability“ von Charles J. Colbourn [Col87] und in „Network reliability and algebraic structures“ von Douglas R. Shier [Shi91] zu finden. In [Agr84b] gibt A. Agrawal eine Übersicht über bekannte Verfahren zur Berechnung von Zuverlässigkeiten in Graphen. J. S. Provan beschreibt in [Pro84] ein Verfahren, das die Minimalschnitte in einem Graphen zur Berechnung der s,t-Zuverlässigkeit benutzt. Ein Verfahren, um diese Minimalschnitte mit Hilfe eines Flussalgorithmuses zu finden, ist in [Cur02] angegeben. Weitere Verfahren zum Finden der Minimalschnitte mit Hilfe einer regulären Algebra [Mar74] und der iterativen Erzeugung [Aru79], [Shi86], [Yeh10]. Zur Berechnung der s,t-Zuverlässigkeit können auch die minimalen s,t-Wege eines Graphen verwendet werden. Entsprechende Verfahren stehen in [Yeh07] und [Col87]. Mit möglichen Reduktionen in gerichteten Graphen beschäftigt sich L. B. Page in [Pag89] und W. W. Bein in [Bei92]. A. Agrawal und A. Satyanarayana stellen in [Agr84a] Untersuchungen über BSP-Graphen (engl.: basically-series-parallel) an, mit deren Hilfe die Zuverlässigkeit in einer großen Klasse von Graphen berechnet werden kann. In letzter Zeit sind mehrere Verfahren aufgekommen, die gerichtete azyklische Graphen (engl.: Propositional Directed Acyclic Graphs) zur Berechnung der s,t-Zuverlässigkeit verwenden. Eine Einführung in diese Herangehensweise ist in [Jon07] und [Jon08] zu finden.

Auf Graphen mit beschränkter Baum- beziehungsweise Wegweite lassen sich viele Probleme der Graphentheorie in polynomialer Zeit lösen. Auch für die Berechnung der Zuverlässigkeit können Algorithmen verwendet werden, die eine Weg- beziehungsweise Baumzerlegung des Graphen verwenden. Entsprechende Verfahren für ungerichtete

Graphen sind in [Poe01], [Wol02], [Bei12] und [Luc00] zu finden. Der Aufwand dieser Algorithmen wächst nur linear in der Anzahl der Knoten und Kanten, allerdings exponentiell in der Weg- beziehungsweise Baumweite des Graphen. Deshalb sind diese Verfahren nur für Graphen mit einer geringer Weg- oder Baumweite geeignet. In den letzten Jahren wurden verschiedene Verfahren zur Berechnung der Weg- beziehungsweise Baumweite eines Graphen entwickelt. Diese Verfahren liefern meist zur Weite auch gleich eine passende Zerlegung des Graphen. Bodlaender und Kloks stellten 1991 [Bod91] einen Algorithmus vor, der für ein festes k in $O(n \log^2 n)$ Schritten überprüft, ob ein Graph eine Weg- beziehungsweise Baumweite von k hat. Weitere Verfahren wurden von K. Cattell [Cat96] und V. Gogate [Gog04] genannt. H. Tamaki veröffentlichte 2011 einen Artikel [Tam11], in dem er einen Algorithmus anführt, der in $O(mn^{k+1})$ Schritten überprüft, ob ein gerichteter Graph die Wegweite k hat. Falls dies der Fall ist, liefert der Algorithmus auch eine entsprechende Wegzerlegung. In [Bod98] gibt H. L. Bodlaender zudem einen Überblick über verschiedene Eigenschaften von Graphen mit beschränkter Weg- beziehungsweise Baumweite und nennt Schranken für die Wegweite in speziellen Graphen.

In der Praxis sind Verfahren, welche die K-Terminal-Zuverlässigkeit berechnen, wichtig. Eine Einführung in diese Verfahren ist in [Luc97], [Woo85] und [Woo89] zu finden. Für spezielle Graphenklassen, wie Serien-Parallele-Graphen [Sat85] und Permutations- und Intervallgraphen [Abo90], wurden besondere Algorithmen entwickelt. Die Berechnung der s,T-Zuverlässigkeit in azyklischen Graphen wurde in [Kon99] näher untersucht. Die Berechnung der Erreichbarkeit wird in den Artikeln von J. I. Brown [Bro05] und [Bro09] und A. Satyanarayana [Sat81] näher analysiert. Hier wird auch die Berechnung in speziellen Graphenklassen thematisiert. M.O. Ball beschäftigt sich in [BP83] mit der Berechnung von Schranken für die Erreichbarkeit und Zusammenhangswahrscheinlichkeit in gerichteten, beziehungsweise ungerichteten Graphen.

3 s,t-Zuverlässigkeit

In diesem Kapitel wird die s,t-Zuverlässigkeit $Conn_{st}(G)$ in gerichteten Graphen betrachtet. Dabei werden zunächst im Abschnitt 3.1 Reduktionen vorgestellt, mit denen die Größe des Graphen reduziert werden kann. Der folgende Abschnitt 3.2 zeigt zwei Möglichkeiten auf, wie ein Graph mittels einer Artikulation oder einer Brücke in zwei kleinere Teilgraphen zerlegt werden kann. Im Abschnitt 3.3 werden exakte Algorithmen für die Berechnung der s,t-Zuverlässigkeit vorgestellt. Im Abschnitt 3.4 wird die Berechnung der s,t-Zuverlässigkeit in speziellen Graphenklassen untersucht.

3.1 Reduktionen

Um die Größe eines Graphen vor der Berechnung der Zuverlässigkeit zu reduzieren, ist es möglich, verschiedene Reduktionen auszuführen. Im ungerichteten Fall sind eine Vielzahl von Reduzierungen bekannt. Die einfachsten sind die Serien- und Parallelreduktion. Diese beiden Reduktionen sind auch im gerichteten Fall möglich. Zusätzlich kann hier auch noch eine Verbindung beider Reduktionen, die Serien-Parallel-Reduktion, eingeführt werden. In Abbildung 3.1 sind diese drei Reduktionen zu sehen. Bei der Parallelreduktion wird ausgenutzt, dass die Bögen unabhängig voneinander ausfallen. Somit können die parallelen Bögen durch eine einzelne ersetzt werden. Bei der Serienreduktion werden die beiden Bögen ebenfalls durch eine einzelne ersetzt und die neue Ausfallwahrscheinlichkeit wird aus dem Produkt der alten berechnet. Die Serien-Parallel-Reduktion ist eine Verallgemeinerung der Serienreduktion.

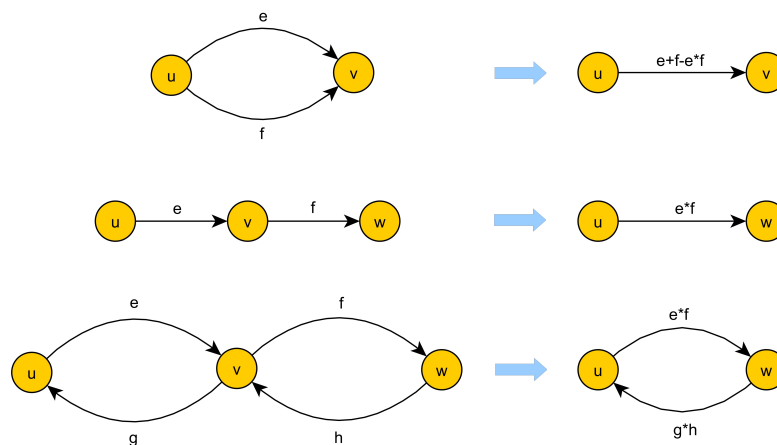


Abbildung 3.1: Parallel-, Serien- und Serien-Parallel-Reduktion

Die Serien- und die Serien-Parallel-Reduktion sind ausschließlich mit Knoten möglich, die zwei Nachbarn im Graphen besitzen. In Abbildung 3.2 sind die sechs möglichen Fälle zu sehen, wie solche Knoten im Graphen auftreten können. Der erste Fall kann

durch eine Serien-Parallel-Reduktion und der vierte Fall durch eine Serienreduktion abgehandelt werden. Im zweiten Fall kann der Bogen f gelöscht werden, da der Knoten v nur über den Knoten u erreichbar ist und somit der Bogen $f = (v, u)$ überflüssig ist. Anschließend kann eine Serienreduktion durchgeführt werden. Im dritten Fall kann analog zum vorherigen Fall der Bogen e gelöscht und anschließend eine Serienreduktion durchgeführt werden. Im fünften und sechsten Fall führt kein Bogen aus dem Knoten v heraus, beziehungsweise hinein. Der Knoten kann somit gelöscht werden.

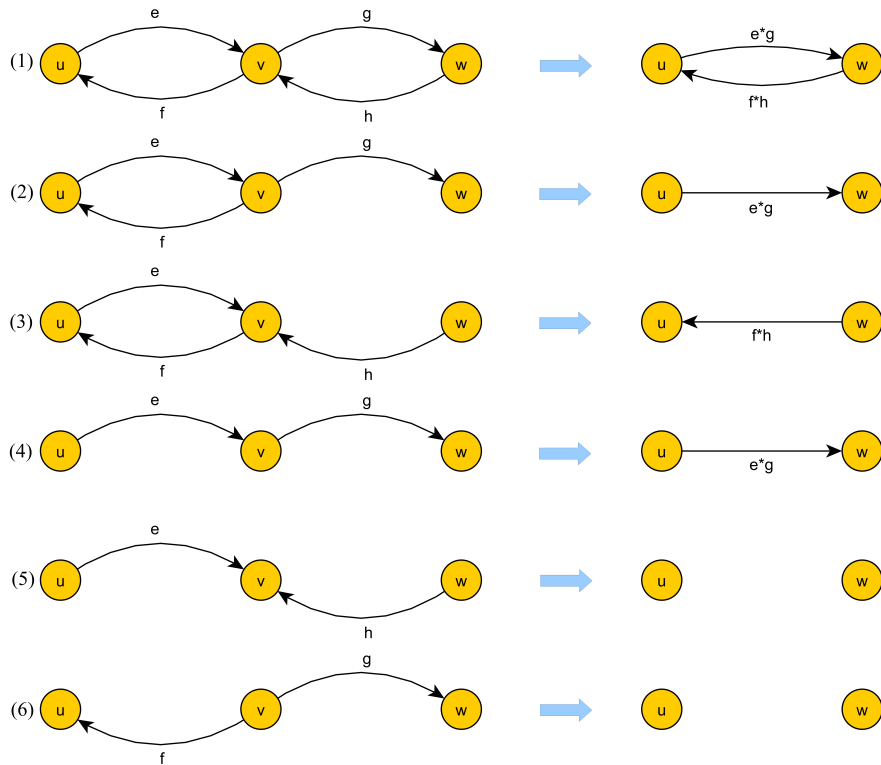


Abbildung 3.2: Arten von Knoten mit zwei Nachbarn

Die Reduktion von nicht benötigten Bögen kann auch auf Knoten mit mehr als zwei Nachbarn angewandt werden. Diese Bogenreduktionen treten bei Knoten mit dem Innenbeziehungsweise Außengrad Eins auf. In der Abbildung 3.3 sind zwei Kantenreduktionen bei Knoten mit drei Nachbarn zu sehen. Dabei darf der Knoten x weder eine Quelle noch eine Senke sein. Im Fall (a) wird der Knoten x nur über den Knoten v erreicht. Zum Knoten v darf nicht mehr zurückgekehrt werden. Somit kann der Bogen e entfernt werden. Im Fall (b) kann der Knoten x nur zum Knoten v verlassen werden. Damit kann der Bogen f gelöscht werden.

Der Ausgangsgraph lässt sich noch weiter verkleinern, indem für die Zuverlässigkeitsberechnung nicht benötigte Knoten weggelassen werden. Dies sind alle Knoten, die nicht auf einem gerichteten s-t-Weg liegen. Sie tragen somit nicht zu der s,t-Zuverlässigkeit bei und können gelöscht werden. Auch alle Bögen, die in den Knoten s hineinführen, und alle Bögen, die aus dem Knoten t herausführen, werden für die Zuverlässigkeitsberechnung nicht benötigt und können gelöscht werden. In Abbildung 3.4 sind diese

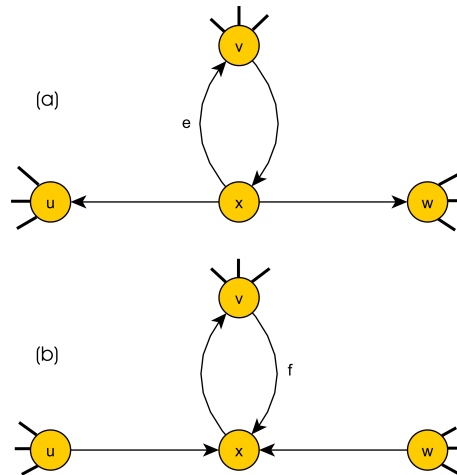


Abbildung 3.3: Kanten-Reduktion

Reduktionsmöglichkeiten zu sehen. Die Knoten 5 und 7 können für die Berechnung weggelassen werden, da sie nicht auf einem 1,8-Weg liegen. Auch der Bogen (8,2) kann gelöscht werden, da er aus der Senke hinausführt.

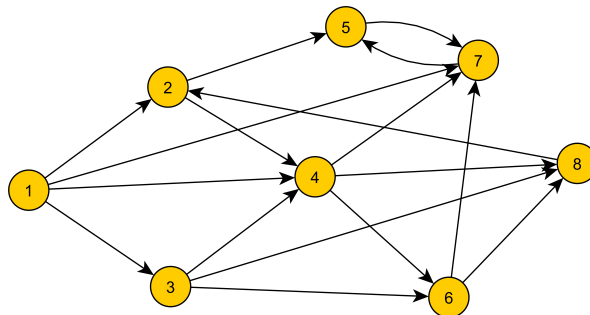


Abbildung 3.4: Reduktion durch Entfernen unnötiger Knoten und Bögen

Die bisher vorgestellten Reduktionen sind zuverlässigkeitserhaltende Reduktionen. Es gilt somit $R_{st}(G) = R_{st}(G')$, wobei G der Ausgangsgraph und G' der Graph ist, der durch die Reduktionen aus G entsteht. Es gibt auch Reduktionen, die nicht zuverlässigkeitserhaltend sind. Bei diesen Reduktionen gilt: $R_{st}(G) = \Omega_i R_{st}(G')$. Eine dieser möglichen Reduktionen ist die s-Dreiecks-Reduktion. Diese Reduktion kann angewandt werden, wenn der Knoten s zwei Nachbarknoten besitzt, die entweder durch einen oder durch zwei Bögen verbunden sind. Bei der s-Dreiecks-Reduktion werden diese Verbindungsbögen weggelassen und die Wahrscheinlichkeiten für die zwei verbleibenden Bögen neu berechnet. Die zwei möglichen Typen sind in Abbildung 3.5 abgebildet. Der folgende Satz trifft eine Aussage über die Berechnung der neuen Kantengewichte und des Faktors Ω_i .

Satz 3.1 $G = (V, E)$ sei ein gerichteter Graph mit einem s-Dreieck vom Typ 1 oder Typ 2. G' sei der Graph, der aus G durch Anwendung der s-Dreiecks-Reduktion entsteht.

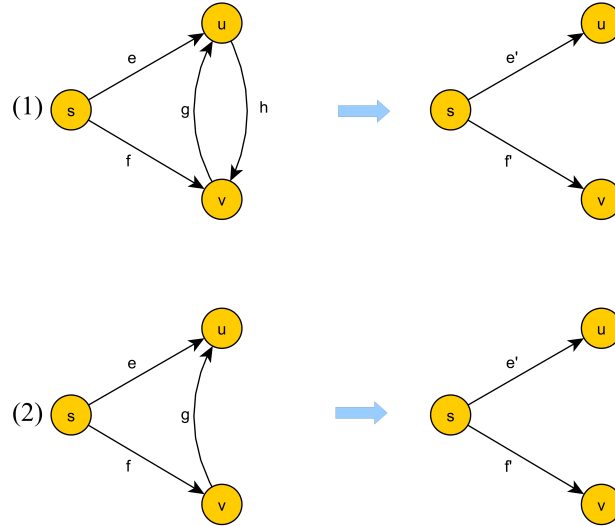


Abbildung 3.5: s-Dreiecks-Reduktion

Dann gelten folgende Beziehungen für die Bogengewichte in G' :

Typ 1:

- $p_{e'} = \frac{p_e(p_f + q_f p_h) + q_e p_f p_g}{p_f + p_e q_f p_h} = \frac{p(-3+2p)}{-1-p+p^2}$
- $p_{f'} = \frac{p_e(p_f + q_f p_h) + q_e p_f p_g}{p_e + q_e p_f p_g} = \frac{p(-3+2p)}{-1-p+p^2}$
- $\Omega_1 = \frac{(p_f + p_e q_f p_h)(p_e + q_e p_f p_g)}{p_e(p_f + q_f p_h) + q_e p_f p_g} = -\frac{(1+p-p^2)^2}{-3+2p}$

Typ 2:

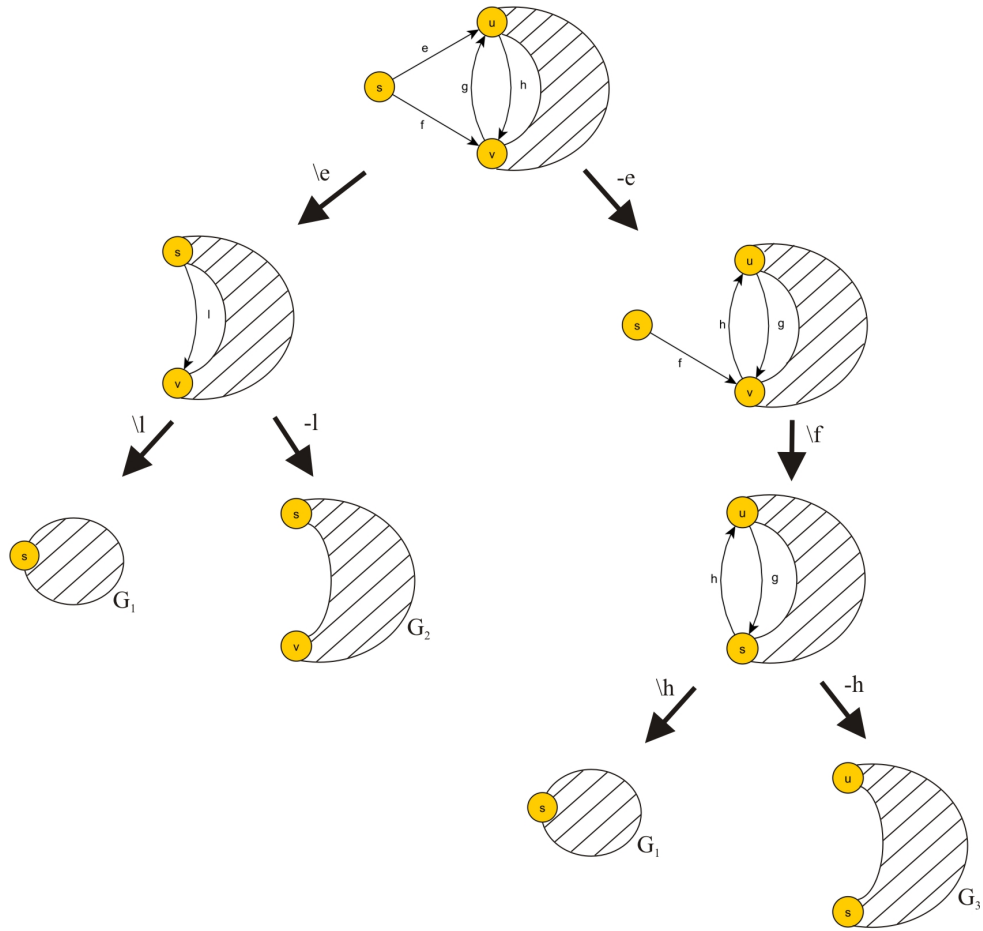
- $p_{e'} = p_e + q_e p_g = 2p - p^2$
- $p_{f'} = \frac{p_e p_f + q_e p_f p_g}{p_e + q_e p_f p_g} = \frac{2p-p^2}{1+p-p^2}$
- $\Omega_2 = \frac{p_e + q_e p_f p_g}{p_e + q_e p_g} = \frac{1+p-p^2}{2-p}$

Für den Graphen G und den Graphen G' gilt mit diesen Bogengewichten die Gleichung $R_{st}(G) = \Omega_i R_{st}(G')$.

Beweis: Der Typ 2 ist nur ein Spezialfall von Typ 1, bei dem der Bogen h eine Verfügbarkeitswahrscheinlichkeit von 0 besitzt. Es genügt somit, die Formeln für den Typ 1 zu beweisen. Der Beweis wird mit Hilfe der Dekompositionsformel, die in Abschnitt 3.3.3 vorgestellt wird, geführt:

$$R_{st}(G) = (1 - p_e)R_{st}(G - e) + p_e R_{st}(G/e) \quad (3.1)$$

Zunächst wird für den Ausgangsgraphen die Zuverlässigkeit $R_{st}(G)$ ermittelt. Der Dekompositionsbaum für diesen Graphen ist in Abbildung 3.6 zu sehen.


 Abbildung 3.6: Dekompositionsbaum für den Graphen G

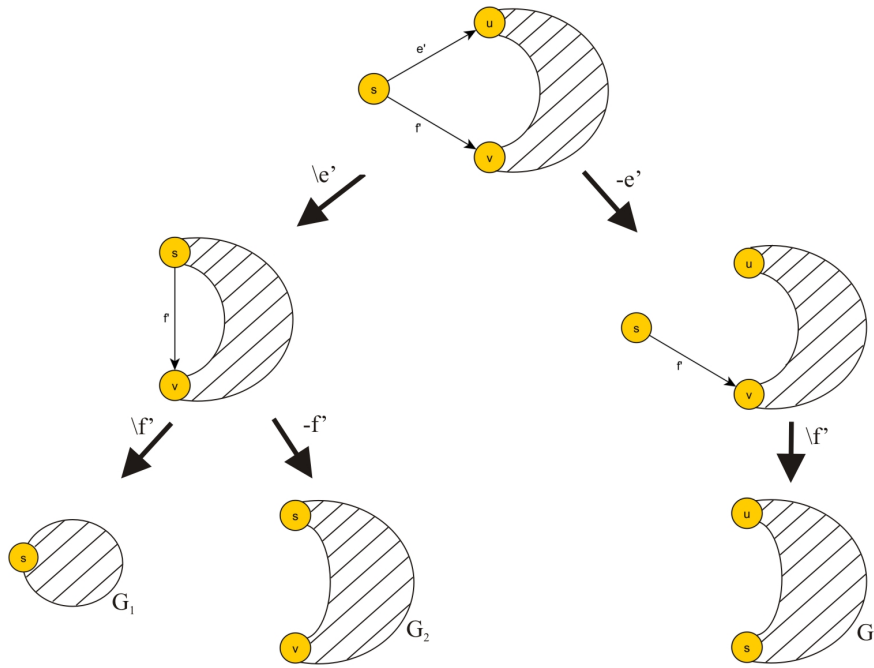
Aus diesem Dekompositionsbaum lässt sich die s,t-Zuverlässigkeit für den Graphen G in Abhängigkeit von der s,t-Zuverlässigkeit der drei Restgraphen G_1, G_2 und G_3 ablesen. In dieser Gleichung gilt: $p_l = p_f + p_g - p_f p_g$.

$$\begin{aligned} R_{st}(G) &= p_e(p_l G_1 + q_l G_2) + q_e p_f(p_h G_1 + q_h G_3) \\ &= (p_e p_l + q_e p_f p_h) G_1 + p_e q_l G_2 + q_e p_f q_h G_3 \end{aligned} \quad (3.2)$$

Analog wird die Dekompositionsformel auf den Graphen G' angewendet. Der so entstehende Dekompositionsbaum ist in Abbildung 3.7 zu sehen. Aus diesem Dekompositionsbaum ergibt sich folgende s,t-Zuverlässigkeit für den Graphen G' in Abhängigkeit von den drei Restgraphen G_1, G_2 und G_3 :

$$\begin{aligned} R_{st}(G') &= p_{e'}(p_{f'} G_1 + q_{f'} G_2) + q_{e'} p_{f'} G_3 \\ &= p_{e'} p_{f'} G_1 + p_{e'} q_{f'} G_2 + q_{e'} p_{f'} G_3 \end{aligned} \quad (3.3)$$

Aus den zwei Gleichungen (3.2) und (3.3) wird nun ein Gleichungssystem zur Ermittlung

Abbildung 3.7: Dekompositionsbaum für den Graph G'

der Wahrscheinlichkeiten $p_{e'}$ und $p_{f'}$ und des Faktors Ω aufgestellt.

$$\begin{aligned} p_e p_l + q_e p_f p_h &= \Omega p_{e'} p_{f'} \\ p_e q_l &= \Omega p_{e'} q_{f'} \\ q_e p_f q_h &= \Omega q_{e'} p_{f'} \end{aligned}$$

Das Lösen dieses Gleichungssystems führt auf die im Satz angegebene Berechnung der neuen Wahrscheinlichkeiten $p_{e'}$ und $p_{f'}$ und des Faktors Ω . \square

Folgerung 3.2 Die *s-Dreiecks-Reduktion* kann auch auf den Knoten t angewandt werden (siehe Abbildung 3.8).

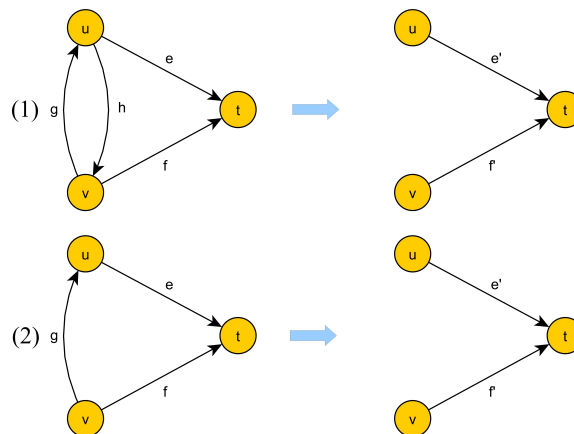


Abbildung 3.8: t-Dreiecks-Reduktionen

3.2 Artikulationen und Brücken

Falls in einem Graphen eine Artikulation oder eine Brücke existiert, kann der Graph für die Berechnung der Zuverlässigkeit in zwei kleinere Teilgraphen aufgeteilt werden. Zunächst erfolgt die Definition einer Artikulation in einem Graphen.

Definition 3.3 $G = (V, E)$ sei ein gerichteter, zusammenhängender Graph und $v \in V$ ein Knoten des Graphen. Der Knoten v heißt Artikulation, falls der Graph G durch Entfernen des Knotens v in zwei Komponenten zerfällt.

Eine solche Artikulation in einem Graphen ist in Abbildung 3.9 zu sehen. Im Untergraphen G wird nun die s,v-Zuverlässigkeit $R_{sv}(G)$ und im Untergraphen H die v,t-Zuverlässigkeit $R_{vt}(H)$ berechnet. Anschließend werden beide Wahrscheinlichkeiten zu der s,t-Zuverlässigkeit multipliziert: $R_{st} = R_{sv}(G)R_{vt}(H)$.

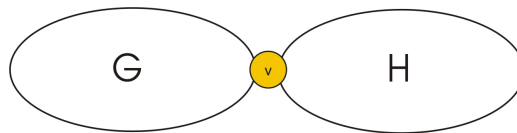


Abbildung 3.9: Artikulation

R. Sedgwick [Sed91] hat 1991 einen Algorithmus vorgestellt, der mit Hilfe der Tiefensuche eine Artikulation in einem Graphen findet. Die Idee hinter diesem Verfahren ist, dass, wenn ein Knoten eine Artikulation ist, dann existieren im Tiefensuchbaum (DFS-Baum) keine Rückwärtskanten von einem tiefer liegenden zu einem höher liegenden Knoten. Würde eine solche Kante existieren, zerfiel der Graph beim Entfernen des Knotens nicht in zwei Teilgraphen. Der Startknoten in diesem Algorithmus muss gesondert betrachtet werden, da er keinen höher liegenden Knoten besitzt. Es gilt: Der Wurzelknoten des DFS-Baumes ist eine Artikulation, wenn er mehr als ein Kind im DFS-Baum hat. Dies gilt, da alle Knoten, die einen Weg zum ersten Kind besitzen, der nicht den Wurzelknoten enthält, unterhalb des Kindes auftauchen. Alle Kinder des Wurzelknotens sind somit nur durch Wege verbunden, die den Wurzelknoten enthalten. Im Algorithmus wird im Vektor *pre_order* die Besuchsreihenfolge gespeichert. In m wird die kleinste *pre_order*-Nummer eines Knotens gespeichert, der von einem Kind aus erreichbar ist. Ist diese Zahl m größer als die *pre_order*-Nummer des aktuellen Knotens v , dann existiert keine Verbindung von den Kindern des Knotens zu einem Knoten oberhalb von v . Somit ist v eine Artikulation.

Der Aufwand zum Finden einer Artikulation ist $O(n + m)$, da jeder Knoten einmal und jede Kante zweimal betrachtet werden muss.

In Abbildung 3.10 wird ein Graph mit einer Brücke dargestellt. Hier wird zunächst wieder die s,v-Zuverlässigkeit im Graphen G und anschließend die w,t-Zuverlässigkeit im Graphen H berechnet. Um die s,t-Zuverlässigkeit zu bestimmen, werden nun die beiden

Algorithmus 1 visit(G,v) [Sed91]**Eingabe:** Graph G , Knoten v

```

v.id = v.id + 1
pre_order[v] = v.id
min_order = v.id
for jeden Knoten  $w$  der adjazent zu  $v$  ist do
    if pre_order[w] == 0 then
        if  $v == \text{startpoint}$  then
            startcount++
        end if
         $m = \text{visit}(g,w)$ 
        if  $m < \text{min\_order}$  then
            min_order = m
        end if
        if  $m \geq \text{pre\_order}[v]$  and  $v \neq \text{startpoint}$  then
             $v$  ist Artikulation
        end if

    else if pre_order[w] < min_order then
        min_order = pre_order[w]
    end if
end for
Ausgabe: min_order

```

ermittelten Wahrscheinlichkeiten und die der Kante (v, w) miteinander multipliziert.

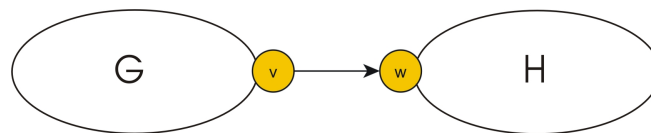


Abbildung 3.10: Brücke

Die Teilung eines Graphen an einer trennenden Knotenmenge X ist eine Verallgemeinerung der Trennung durch Artikulationen. In gerichteten Netzwerken muss aber beachtet werden, dass Wege von s nach t die trennende Knotenmenge X mehrfach passieren können (siehe Abbildung 3.11). Dadurch ist es sehr schwer, die s,t-Zuverlässigkeit zu bestimmen. Ein einfacherer Fall tritt auf, wenn der Graph azyklisch ist. Hier kann nur genau ein Knoten der Menge X in einem s,t-Weg vorkommen. Dieser Fall wird in Abschnitt 3.4.9 eingehender betrachtet.

3.3 Exakte Algorithmen

In diesem Abschnitt werden verschiedene exakte Algorithmen für die Berechnung der s,t-Zuverlässigkeit vorgestellt. Alle hier vorgestellten Algorithmen haben, auf Grund der

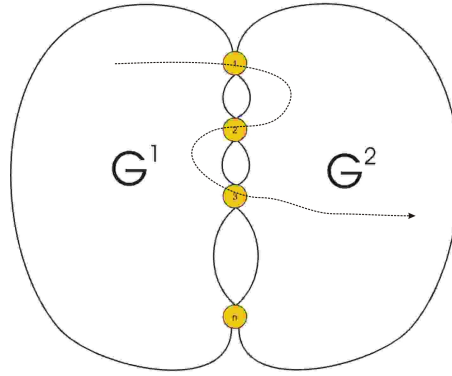


Abbildung 3.11: Mehrfaches passieren der trennenden Knotenmenge

Komplexität des s,t-Zuverlässigkeitsproblems, einen exponentiellen Aufwand. Die einfachste Methode, die Zuverlässigkeit zu berechnen, ist die vollständige Enumeration aller möglichen Zustände. In den nachfolgenden Abschnitten wird die Berechnung mit dem Inklusion-Exklusion-Prinzip (3.3.2) und mit Hilfe der Dekompositionsformel (3.3.3) vorgestellt. Anschließend wird eine Berechnungsmethode mit Hilfe von Schnitten (3.3.4) erläutert.

3.3.1 Vollständige Enumeration

Die einfachste Methode um die s,t-Zuverlässigkeit $Conn_2$ zu berechnen, ist die vollständige Enumeration. Sie wurde in diesem Zusammenhang erstmalig von Moore und Shannon [Moo56] beschrieben. Die Idee ist, dass alle Bögen des Graphen G zwei mögliche Zustände haben, intakt oder ausgefallen. Der Status des Graphen kann mit einem 0-1-Vektor $\delta = (\delta_1, \delta_2, \dots, \delta_m)$ dargestellt werden. Dabei gilt $\delta_i = 1$, falls der Bogen i intakt ist und andernfalls 0. Die Wahrscheinlichkeit für einen Zustand δ kann somit durch

$$P(\delta) = \prod_{k=1}^m p_k^{\delta_k} (1 - p_k)^{1-\delta_k}$$

angegeben werden. Um die s,t-Zuverlässigkeit mit den soeben eingeführten Zuständen angeben zu können, wird noch die 0,1-Variable $\phi_{s,t}(\delta)$ benötigt. Sie ist genau dann 1, wenn im Untergraphen der intakten Bögen, also der Bögen mit $\delta_i = 1$, ein Weg von s nach t existiert. \mathcal{D} sei die Menge aller möglichen Zustände des Graphen. Mit Hilfe dieser Variablen und den Statuswahrscheinlichkeiten kann nun folgende Formel für die s,t-Zuverlässigkeit aufgestellt werden:

$$R_{st}(G) = \sum_{\delta \in \mathcal{D}} \phi_{s,t}(\delta) P(\delta) \quad (3.4)$$

In der Praxis ist dieses Verfahren nicht zu verwenden, da für die Größe der Zustandsmenge \mathcal{D} , $|\mathcal{D}| = 2^m$ gilt und für jeden Zustand geprüft werden müsste, ob ein s,t-Weg

existiert.

Beispiel 3.4 Für den in Abbildung 3.12 zu sehenden Graphen soll die 1,4-Zuverlässigkeit berechnet werden. Alle Bögen sind mit der Zuverlässigkeit $p_e = p = 0.8$ gewichtet.

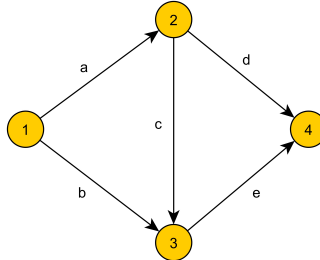


Abbildung 3.12: Graph mit 4 Knoten

Bei diesem Graphen ist es klar, dass das System funktionsfähig ist, falls höchstens ein Bogen ausfällt oder zwei Bögen ausfallen, außer den Bögen $\{a, b\}$, $\{a, e\}$ und $\{d, e\}$. Das System fällt aus, falls drei oder mehr Bögen ausfallen. Außer die Bögen $\{a, c, d\}$ oder die Bögen $\{b, c, e\}$ sind ausgefallen. Mit diesen Informationen kann nun das Zuverlässigkeitspolynom aufgeschrieben und anschließend $q = 1 - p$ im Polynom ersetzt werden.

$$\begin{aligned}
 R_{st}(G) &= p^5 + 5p^4q + 7p^3q^2 + 2p^2q^3 \\
 &= p^5 - 3p^4 + p^3 + 2p^2 \\
 &= 0.89088
 \end{aligned}$$

3.3.2 Inklusion-Exklusion

In diesem Abschnitt wird die Anwendung des Inklusion-Exklusion-Prinzips zur Berechnung der s,t-Zuverlässigkeit vorgestellt. In der im vorherigen Abschnitt vorgestellten Gleichung (3.4) werden nur die Summanden benötigt, für die $\phi_{s,t}(\delta) = 1$ gilt. Dies ist genau dann der Fall, wenn es einen s,t-Weg im induzierten Untergraphen gibt. Es liegt somit nahe, sich von vornherein auf die s,t-Wege im Graphen G zu beschränken. $\{P_1, P_2, \dots, P_K\}$ sei die Menge der minimalen s,t-Wege in G . Sei E_i das Ereignis, dass alle Bögen im Weg P_i intakt sind. Die s,t-Zuverlässigkeit ist somit die Wahrscheinlichkeit der Vereinigung der Ereignisse E_i . Es gilt also:

$$R_{st} = P(E_1 \cup E_2 \cup \dots \cup E_k) \quad (3.5)$$

Das Inklusion-Exklusion-Prinzip kann für die Berechnung der Wahrscheinlichkeit von nicht disjunkten Ereignissen verwendet werden. Der folgende Satz beschreibt dieses

Prinzip.

Satz 3.5 Seien A_i , für $i = 1, \dots, n$, Ereignisse. Für die Wahrscheinlichkeit der Vereinigung dieser Ereignisse A_i gilt:

$$P\left(\bigcup_{i=1}^n A_i\right) = \sum_{k=1}^n (-1)^{k+1} \sum_{\substack{I \subseteq \{1, \dots, n\} \\ |I|=k}} P\left(\bigcap_{i \in I} A_i\right)$$

Mit Hilfe des Satzes 3.5 kann die Gleichung (3.5) wie folgt geschrieben werden:

$$R_{st}(G) = \sum_i P(E_i) - \sum_{i < j} P(E_i E_j) + \sum_{i < j < l} P(E_i E_j E_l) - \dots + (-1)^{k+1} P(E_1 \dots E_k) \quad (3.6)$$

Die Anzahl der Summanden in dieser Formel wächst exponentiell zur Anzahl der s,t-Wege. Sei k die Anzahl der s,t-Wege, dann hat die Summe $2^k - 1$ Terme. Für große Graphen ist somit auch dieses Verfahren nicht praktikabel.

Beispiel 3.6 Der Graph in Abbildung 3.12 hat 3 verschiedene minimale s,t-Wege:

$$P_1: (a, d), P_2: (b, e), P_3: (a, c, e)$$

Jetzt müssen nur noch die Wahrscheinlichkeiten für die einzelnen Ereignisse ausgerechnet werden. So gilt zum Beispiel $P(E_1) = p_a p_d$ und $P(E_1 E_2) = p_a p_b p_d p_e$. Das Einsetzen der einzelnen Wahrscheinlichkeiten in die Formel (3.6) liefert:

$$\begin{aligned} R_{st}(G) &= (p_a p_d + p_b p_e + p_a p_c p_e) - (p_a p_b p_d p_e + p_a p_c p_d p_e + p_a p_b p_c p_e) \\ &\quad + p_a p_b p_c p_d p_e \\ &= p_a p_d + p_b p_e - p_a p_e (-p_c + p_b p_d + p_c p_d + p_b p_c - p_b p_c p_d) \end{aligned}$$

Das Ersetzen von p_i durch p und das Zusammenfassen der Terme ergibt wieder das bekannte Zuverlässigkeitspolynom:

$$\begin{aligned} R_{st}(G) &= p^2 + p^2 - p^2(-p + p^2 + p^2 + p^2 - p^3) \\ &= p^5 - 3p^4 + p^3 + 2p^2 \end{aligned}$$

3.3.3 Dekomposition

Eine andere Möglichkeit, die vollständige Enumeration zu betrachten, ist den Zustand eines einzelnen Bogens anzusehen und das Problem rekursiv in zwei kleinere Teilprobleme zu zerlegen. Ein Bogen $e \in E$ kann dabei ausgefallen sein, dann wird nur noch

der Graph $G - e$ betrachtet, oder er kann intakt sein. In diesem Fall wird der Graph G/e , in dem die Kante e kontrahiert wurde, betrachtet. Diese Überlegung führt zu folgender Formel, die zuerst von Moskowitz [Mos] beschrieben wurde.

$$R_{st}(G) = (1 - p_e)R_{st}(G - e) + p_e R_{st}(G/e) \quad (3.7)$$

In jedem entstehenden Teilgraphen werden zunächst Serien- und Parallelreduktionen ausgeführt und anschließend wird wieder die Dekompositionsformel angewendet. In jedem der entstehenden Untergraphen sollte überprüft werden, ob überhaupt noch ein s,t-Weg vorhanden ist. Ist keiner mehr vorhanden, muss dieser Untergraph nicht weiter untersucht werden. Durch die Anwendung der Reduktionen und der Überprüfung auf s,t-Wege, kann die Anzahl der Rekursionsschritte erheblich reduziert werden. Werden keine Reduktionen und vorzeitige Abbrüche durchgeführt, führt diese Formel wieder auf eine vollständige Enumeration aller möglichen Zustände des Graphen.

In einem ungerichteten Graphen spielt die Auswahl der Kanten für die Dekompositionsformel allenfalls eine algorithmische Rolle. In gerichteten Graphen ist die Auswahl der Bögen für die Korrektheit der Dekompositionsformel von Bedeutung. Hier dürfen nur Bögen verwendet werden, die inzident zum Knoten s oder zum Knoten t sind. Wird ein anderer Bogen gewählt, können bei der Kontraktion des Bogens Wege im Graphen entstehen, die zuvor nicht vorhanden waren. Dieser Sachverhalt ist in Abbildung 3.13 zu sehen. Nach der Kontraktion des Bogens $(4, 2)$ entsteht ein neuer Weg $1, 2, 5$, der im Ursprungsgraphen nicht vorhanden war. Dieses Problem tritt bei Verwendung von Bögen, die aus der Quelle s herausgehen, beziehungsweise in die Senke t hineingehen, nicht auf. Durch eine Kontraktion dieser Bögen entstehen keine neuen s,t-Wege im Graphen.

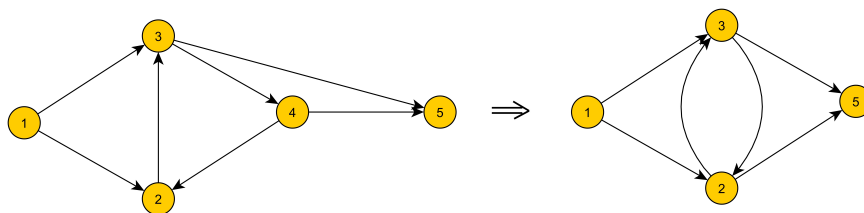


Abbildung 3.13: Problem bei der Kontraktion eines Bogens

Beispiel 3.7 Die Anwendung der Dekompositionsmethode wird wieder anhand des Graphen aus Abbildung 3.12 veranschaulicht. In Abbildung 3.14 sind die einzelnen Dekompositionsschritte zu sehen. In jedem Schritt wurden, falls möglich, parallele Bögen reduziert und Schlingen entfernt. Die Dekomposition kann abgebrochen werden, falls nur noch ein s,t-Weg im Graphen existiert.

Jetzt müssen nur noch die ermittelten Zuverlässigkeiten aus dem Baum abgelesen und

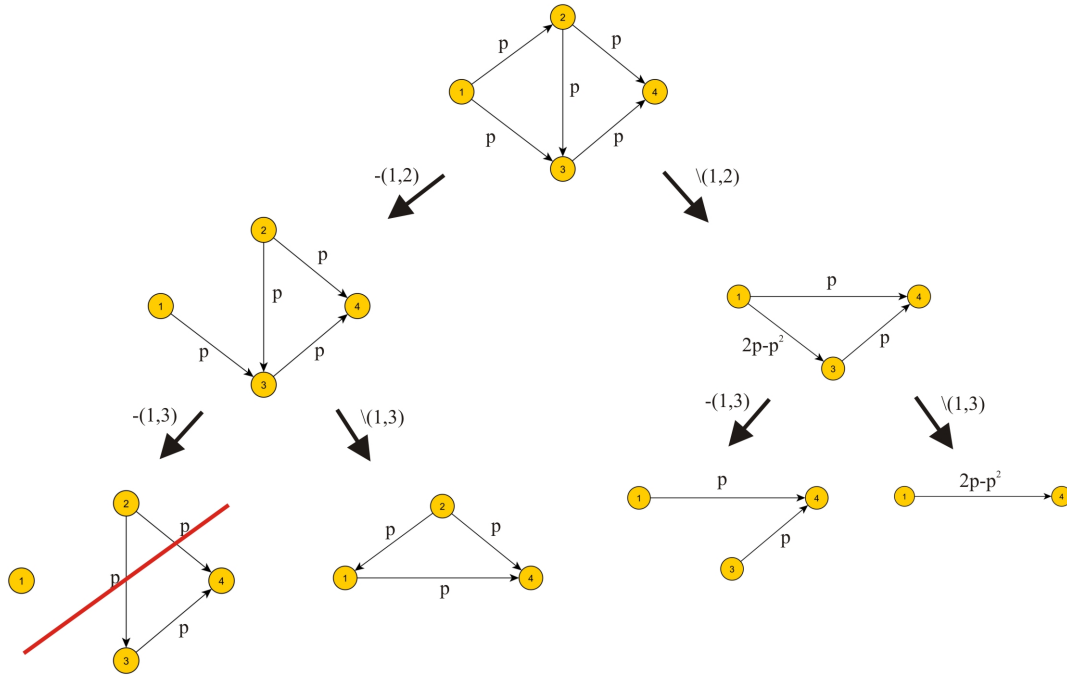


Abbildung 3.14: Dekompositionsbaum

zusammengefasst werden.

$$\begin{aligned} R_{st}(G) &= (1-p)p^2 + p((1-(2p-p^2))p + (2p-p^2)(2p-p^2)) \\ &= p^5 - 3p^4 + p^3 + 2p^2 \end{aligned}$$

3.3.4 Berechnung mit Hilfe von Schnitten

In Abschnitt 3.3.2 wurde gezeigt, wie mit Hilfe der minimalen Wege des Graphen die s,t-Zuverlässigkeit berechnet werden kann. In diesem Abschnitt wird nun eine Methode vorgestellt, mit der die s,t-Zuverlässigkeit mit Hilfe von minimalen s,t-Schnitten im Graphen berechnet werden kann. Die Anzahl der s,t-Schnitte in einem Graphen ist im Allgemeinen geringer als die Anzahl der minimalen s,t-Wege. So gibt es in einem Graphen mit n Knoten und m Kanten höchstens 2^{n-2} minimale s,t-Schnitte, wobei die Anzahl der minimalen s,t-Wege durch 2^{mn+1} beschränkt ist. Im Allgemeinen ist es somit sinnvoll, sich zur Berechnung der s,t-Zuverlässigkeit auf die Verwendung von Schnitten, anstatt Wegen, zu konzentrieren. Die Laufzeit des hier vorgestellten Algorithmus ist polynomiell in der Anzahl der Schnitte und wurde von Provan und Ball [Pro84] erstmalig 1984 beschrieben.

Um den Algorithmus beschreiben zu können, werden noch einige grundlegende Definitionen benötigt. Zunächst wird ein s,t-Schnitt C_{st} in einem Graphen definiert.

Definition 3.8 $G = (V, E)$ sei ein gerichteter, zusammenhängender Graph und $s, t \in V$

seien zwei Knoten des Graphen. Ein s,t-Schnitt C_{st} im Graphen G ist eine Teilmenge der Bogenmenge, bei deren Entfernen aus dem Graphen, der resultierende Graph $H = (V, E - C_{st})$ in zwei Komponenten zerfällt und die Knoten s und t in verschiedenen Komponenten liegen. Ein s,t-Schnitt C_{st} heißt minimal, falls er keinen anderen s,t-Schnitt echt enthält. Die Menge $C(s, t)$ enthält alle minimalen s,t-Schnitte des Graphen.

Definition 3.9 Sei $G = (V, E)$ ein gerichteter Graph. Dann ist

$$E(A, B) = \{(a, b) \in E : a \in A, b \in B\} \subseteq E$$

eine Bogenteilmenge, wobei für die Mengen A und B gilt: $A \subset V, B \subset V, A \cap B = \emptyset$.

Mit Hilfe eines s,t-Schnittes C des Graphen können nun die folgenden drei Knotenteilmengen definiert werden:

$$SN(C) = \{u \in V : \text{es existiert ein Weg von } s \text{ nach } u, \text{ der keinen Bogen von } C \text{ enthält}\}$$

$$TN(C) = \{v \in V : \text{es existiert ein Weg von } v \text{ nach } t, \text{ der keinen Bogen von } C \text{ enthält}\}$$

$$SE(C) = \{u \in SN(C) : \text{es existiert ein Bogen } (u, v), \text{ mit } v \in TN(C)\}$$

Für die Mengen $SN(C)$ und $TN(C)$ gilt: $SN(C) \cap TN(C) = \emptyset$, da $SN(C)$ genau die Knoten enthält die mit s in einer Komponente sind und $TN(C)$ die Knoten enthält, die mit t in einer Komponente sind. Außerdem gilt $C = E(SN(C), TN(C))$, da alle Bögen, die zwischen den beiden Komponenten verlaufen, im Schnitt sein müssen und da der Schnitt minimal ist, können im Schnitt nicht mehr Bögen sein.

Sei $S \subset V$, mit $s \in S$ und $t \notin S$, so dass $G(S)$ einen Weg von s zu jedem Knoten von S enthält. T sei eine maximale Teilmenge von $V - S$, so dass $t \in T$ und $G(T)$ einen Weg von jedem Knoten aus T zu t besitzt. Dann ist $CT(S)$ definiert als die Menge aller Bögen zwischen S und T , also $E(S, T)$. $CT(S)$ ist somit der einzige s,t-Schnitt in $E(S, V - S)$. Die soeben eingeführten Mengen werden nun am Graphen aus Abbildung 3.15 verdeutlicht. Falls $S = \{s, 1, 2, 3, 4\}$ ist, dann folgt daraus $T = \{5, 6, t\}$ und $CT(S) = \{(3, 5), (4, 6)\}$. $SN(C) = \{s, 1, 2, 3, 4, 7\}$, $TN(C) = \{5, 6, t\}$ und $SE(C) = \{3, 4\}$.

Sei nun $E(C)$ das Ereignis, dass alle Kanten des Schnittes C ausfallen. Somit kann für jeden Schnitt $C \in C(s, t)$ folgendes Ereignis formuliert werden:

$$EC(C) = R_{s, SE(C)}(G) \cap E(C)$$

= Es existiert ein intakter Weg von s zu allen Knoten von $SE(C)$,
aber zu keinem Knoten von $TN(C)$

Dabei bezeichnet $R_{s, SE(C)}(G)$ die Wahrscheinlichkeit, dass von s aus alle Knoten der Knotenmenge $SE(C)$ erreicht werden. Die Wahrscheinlichkeiten für die Ereignisse $EC(C)$

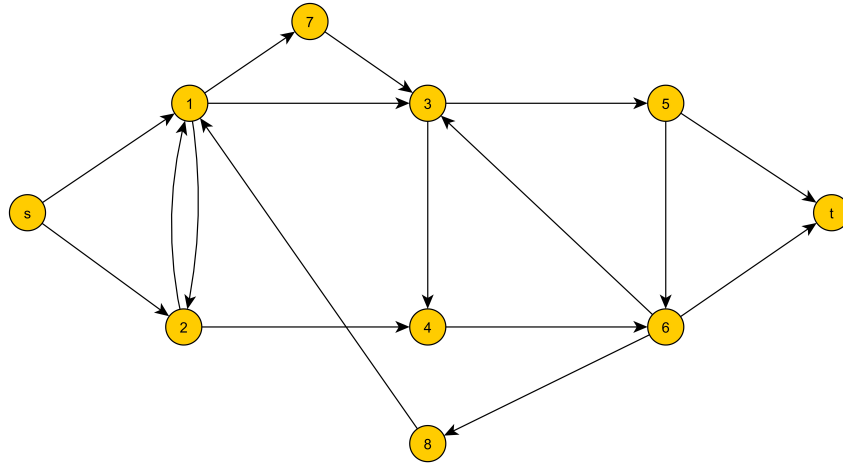


Abbildung 3.15: Beispiel für die Mengen in Schnittalgorithmus

werden jetzt für die Berechnung der s,t-Zusammenhangswahrscheinlichkeit verwendet. Der nächste Satz stellt diese Verbindung her.

Satz 3.10 Für die s,t-Zusammenhangswahrscheinlichkeit gilt:

$$R_{st} = 1 - \sum_{C \in C_{st}} P(EC(C))$$

Beweis: Für den Beweis genügt es zu zeigen, dass

$$\overline{EP(s,t)} = \bigcup_{C \in C(s,t)} EC(C) \quad (3.8)$$

gilt. Dabei ist $EP(s,t)$ das Ereignis, dass es einen intakten s,t-Weg im Graphen gibt. Mit \bigcup wird die disjunkte Vereinigung und mit $\overline{EP(s,t)}$ das Kompliment des Ereignisses $EP(s,t)$ bezeichnet. Um die Gleichung (3.8) zu beweisen, müssen zwei Teilaussagen bewiesen werden:

1. Falls das Ereignis $EC(C)$ eintritt, dann tritt auch $\overline{EP(s,t)}$ ein
2. $\overline{EP(s,t)}$ impliziert, dass genau ein Ereignis $EC(C)$ eintritt

Zu 1.: Falls das Ereignis $EC(C)$ eintritt, dann sind alle Kanten von C ausgefallen und somit kann kein s,t-Weg intakt sein.

Zu 2.: Sei das Ereignis $\overline{EP(s,t)}$ eingetreten und sei

$$S = \{v : EP(s,v) \text{ tritt ein}\} \subseteq V - \{t\}$$

Somit ist $SE(CT(S)) \subseteq S$ und alle Bögen in $CT(S)$ sind ausgefallen. Demnach muss mindestens ein Ereignis $EC(C)$ eintreten und der korrespondierende Schnitt C muss in $E(S, V - S)$ enthalten sein. Da aber $CT(S)$ der einzige Schnitt in $E(S, V - S)$ ist, tritt genau ein Ereignis $EC(C)$ ein. \square

Die Wahrscheinlichkeit $P(EC(C))$ für einen Schnitt C kann in Abhängigkeit von Schnitten C' mit $SN(C') \subset SN(C)$ berechnet werden. Der folgende Satz gibt diesen Zusammenhang wieder.

Satz 3.11 Für jeden Schnitt $C \in C(s, t)$ gilt:

$$P(EC(C)) = \prod_{e \in C} q_e \left(1 - \sum_{\substack{C' \in C(s, t) \\ SN(C') \subset SN(C)}} P(EC(C')) / \prod_{e \in C \cap C'} q_e \right)$$

Beweis: Siehe [Pro84]. □

Unter Verwendung der beiden Sätze kann der Algorithmus 2 zur Berechnung der s,t-Zuverlässigkeit mit Hilfe von Schnitten aufgeschrieben werden.

Algorithmus 2 cut_reliability(G, s, t)

Eingabe: Graph $G=(V,E)$, Startknoten s , Endknoten t

$C(s, t) = \text{generate_st_cuts}(G, s, t)$

for C in $C(s, t)$ **do**

$SN(C) = \{u \in V : \text{Es existiert ein } s, u\text{-Weg der keine Kante aus } C \text{ enthält}\}$

end for

for C in $C(s, t)$: Nach aufsteigender Kardinalität von $SN(C)$ **do**

Berechne $P(EC(C)) = (1 - \sum_{\substack{C' \in C(s, t) \\ \text{mit } SN(C') \subset SN(C)}} [P(EC(C')) / \prod_{e \in C' \cap C} q_e]) \prod_{e \in C} q_e$

end for

Berechne die s,t-Zuverlässigkeit $R_{st} = 1 - \sum_{C \in C(s, t)} P(EC(C))$

Ausgabe: Zuverlässigkeit R_{st}

Die minimalen s,t-Schnitte des Graphen können in $O((m+n)\mu)$ [Tsu80] Schritten berechnet werden, wobei $n = |V|$, $m = |E|$ und $\mu = |C(s, t)|$ gilt. Zur Berechnung der Wahrscheinlichkeiten $P(EC(C))$ muss zunächst die Menge $SN(C)$ gebildet werden. Dies geschieht in $O(n+m)$ Schritten. Die s,t-Schnitte C' können in $O((n+m)\mu')$ Schritten berechnet werden, wobei μ' die Anzahl der Schnitte C' ist. Somit gilt:

$$b = \sum_{C \in C(s, t)} [\text{Anzahl der } C' \in C(s, t) \text{ mit } SN(C') \subset SN(C)] \leq \mu(\mu - 1)/2$$

Hiermit kann die Berechnung der Wahrscheinlichkeiten $P(EC(C))$ in $O((n+m)b)$ Operationen ausgeführt werden. Im letzten Schritt werden noch die einzelnen Wahrscheinlichkeiten aufsummiert. Dies benötigt μ zusätzliche Operationen. Somit ergibt sich als Gesamtaufwand $O((n+m)(b+\mu)) \leq O((n+m)\mu^2)$.

Beispiel 3.12 Gegeben sei wieder der Graph aus Abbildung 3.12. In diesem Graphen

gibt es vier minimale 1,4-Schnitte:

$$S(1,4) = \{\{(1,2), (1,3)\}, \{(1,2), (3,4)\}, \{(1,3), (2,3), (2,4)\}, \{(2,4), (3,4)\}\}$$

Mit Hilfe der Schnitte können jetzt die Mengen $SN(C)$ ermittelt werden:

$$\begin{aligned} SN(\{(1,2), (1,3)\}) &= SN(1) = \{1\} \\ SN(\{(1,2), (3,4)\}) &= SN(2) = \{1, 3\} \\ SN(\{(1,3), (2,3), (2,4)\}) &= SN(3) = \{1, 2\} \\ SN(\{(2,4), (3,4)\}) &= SN(4) = \{1, 2, 3\} \end{aligned}$$

Mit Hilfe der Formel aus Satz 3.11 können nun die Wahrscheinlichkeiten für die Ereignisse $EC(C)$ berechnet werden. Durch einsetzen von $q = 1 - p$ und zusammenfassen ergeben sich folgende Wahrscheinlichkeiten:

$$\begin{aligned} P(EC(1)) &= P(EC(\{(1,2), (1,3)\})) = q^2(1 - (0)) = p^2 - 2p + 1 \\ P(EC(2)) &= P(EC(\{(1,2), (3,4)\})) = q^2(1 - ((p^2 - 2p + 1)/q)) = p^3 - 2p^2 + p \\ P(EC(3)) &= P(EC(\{(1,3), (2,3), (2,4)\})) = q^3(1 - ((p^2 - 2p + 1)/q)) \\ &= -p^4 + 3p^3 - 3p^2 + p \\ P(EC(4)) &= P(EC(\{(2,4), (3,4)\})) \\ &= q^2(1 - (P(EC(1))/q + P(EC(2))/q^2 + P(EC(3))/q^2)) \\ &= -p^5 + 4p^4 - 5p^3 + 2p^2 \end{aligned}$$

Mit Hilfe dieser Polynome und der Formel aus Satz 3.10 kann nun das Zuverlässigkeitspolynom berechnet werden:

$$\begin{aligned} R_{st}(G) &= 1 - \sum_{C \in C_{st}} P(EC(C)) \\ &= 1 - ((p^2 - 2p + 1) + (p^3 - 2p^2 + p) + (-p^4 + 3p^3 - 3p^2 + p) \\ &\quad + (-p^5 + 4p^4 - 5p^3 + 2p^2)) \\ &= p^5 - 3p^4 + p^3 + 2p^2 \end{aligned}$$

Ermittlung aller minimalen s,t-Schnitte

In diesem Abschnitt wird ein Verfahren zur Ermittlung der minimalen s,t-Schnitte in einem gerichteten Graphen vorgestellt. Dieses Verfahren wurde von D. Shier [Shi86] 1986 entwickelt und hat einen quadratischen Aufwand in der Anzahl der Schnitte. Es basiert auf einem Verfahren von A. Martelli [Mar76], bei dem eine algebraische Formulierung der s,t-Schnitte genutzt wird. In der Literatur finden sich noch weitere Verfahren, so zum Beispiel von S. Arunkumar [Aru79], Li-Pu Yeh [Yeh10] oder S. Tsukiyama [Tsu80], die eine lineare Laufzeit in der Anzahl der Schnitte haben und verschiedene Flussalgorithmen

men wie zum Beispiel von N. Curet [Cur02]. Der vorgestellte Algorithmus basiert auf der zugrundeliegenden algebraischen Struktur des Graphen und berechnet gleichzeitig alle minimalen s,v-Schnitte, $\forall v \in V - \{s\}$.

Sei $\mathcal{A} = \{a, b, c, \dots\}$ eine Menge und X eine Menge von Teilmengen von \mathcal{A} . Eine reduzierte Menge $r(X)$ entsteht aus X , indem alle Teilmengen entfernt werden, die echt in einer anderen Teilmenge enthalten sind. Sei \mathcal{S} die Menge aller reduzierten Mengen aus der Menge \mathcal{A} . Für zwei Elemente $X, Y \in \mathcal{S}$ können jetzt die folgenden beiden Operationen definiert werden:

$$X \oplus Y = r(\{\alpha \cup \beta : \alpha \in X, \beta \in Y\}) \quad (3.9)$$

$$X \otimes Y = r(X \cup Y) \quad (3.10)$$

Die Menge \mathcal{S} bildet zusammen mit diesen beiden Operationen eine reguläre Algebra $R = (\mathcal{S}, \oplus, \otimes)$. Die Menge, die nur die leere Menge enthält, ist das Nullelement und das Einselement ist die leere Menge. Mit Hilfe dieser Algebra R kann eine neue Algebra erzeugt werden, die aus allen $n \times n$ -Matrizen besteht, deren Elemente zur Algebra R gehören. Für diese Algebra kann nun eine Addition und eine Multiplikation definiert werden:

$$A + B = [a_{ij} \oplus b_{ij}]$$

$$AB = [\bigoplus_{k=1}^n a_{ik} \otimes b_{kj}]$$

Wobei A und B zwei Matrizen aus dieser Algebra sind. Der Abschluss einer Matrix A ist durch $A^* = \sum_{i=0}^{\infty} A^i$ gegeben. Die Matrix A^* ist eine Lösung der Gleichung

$$Y = AY + E \quad (3.11)$$

Sei nun $G = (V, E)$ ein gerichteter Graph mit n Knoten. A sei eine $n \times n$ -Matrix mit Elementen aus der gerade definierten Algebra K . Das Element a_{ij} der Matrix ist die leere Menge \emptyset , falls kein Bogen zwischen den Knoten i und j existiert, beziehungsweise falls der Bogen existiert, gilt: $a_{ij} = \{(i, j)\}$. Mit Hilfe des Abschlusses A^* der Matrix A , kann nun folgender Satz formuliert werden:

Satz 3.13 [Mar74] *Das Element a_{ij}^* der Matrix A^* gibt die Menge der minimalen i,j-Schnitte im Graphen an.*

Beweis: Es gilt:

$$A^* = I + A + A^2 + A^3 + \dots + A^{n-1}$$

Somit besteht das Element a_{ij}^* aus einer Summe von Wegprodukten aller minimalen

Wege zwischen den Knoten i und j . Somit gilt:

$$a_{ij}^* = \sum_{k=1}^{n-1} a_{ij}^k = \sum_{k=1}^{n-1} \sum_{p \in P_{ij}^k} w(p)$$

Dabei ist P_{ij}^k die Menge aller minimalen Wege zwischen i und j der Länge k und $w(p)$ das Wegeprodukt des Weges p ist. Sei $p = (e_1, e_2, e_3, \dots, e_k)$, dann gilt für das Wegeprodukt $w(p)$:

$$\begin{aligned} w(p) &= \{e_1\} \otimes \{e_2\} \otimes \{e_3\} \otimes \dots \otimes \{e_k\} \\ &= \{e_1, e_2, e_3, \dots, e_k\} \end{aligned}$$

Sei in der Menge M die Addition aller möglichen Wegemengen enthalten. Bei der Addition und auch bei der Bestimmung des Wegeprodukts werden die Obermengen, die entstehen, nicht gelöscht. Durch die Addition enthält jedes Element der Menge M einen Bogen von jedem Weg von i nach j . Somit ist jedes Element von M eine Schnittmenge. Diese Menge enthält sogar alle möglichen Kombinationen von Bögen der minimalen Wege. Somit enthält die Menge M alle möglichen Schnittmengen. Zum Schluss erhält man a_{ij}^* aus M durch Löschen aller Obermengen. Somit enthält a_{ij}^* nur minimale Schnittmengen. \square

Beispiel 3.14 Mit Hilfe der Adjazenzmatrix sollen jetzt alle minimalen 1,4-Schnitte im Graphen in Abbildung 3.12 erzeugt werden. Dazu wird zunächst die Adjazenzmatrix A des Graphen nach der obigen Bildungsvorschrift aufgestellt:

$$A = \begin{pmatrix} \emptyset & \{\{a\}\} & \{\{b\}\} & \emptyset \\ \emptyset & \emptyset & \{\{c\}\} & \{\{d\}\} \\ \emptyset & \emptyset & \emptyset & \{\{e\}\} \\ \emptyset & \emptyset & \emptyset & \emptyset \end{pmatrix}$$

Jetzt können die Potenzen der Adjazenzmatrix gebildet werden, dabei werden nur die für die 1,4-Schnitte benötigten Elemente berechnet:

$$\begin{aligned} A^2 &= \begin{pmatrix} \{\{a,b\}\} & \{\{a,b\}\} & \{\{a,b\}, \{b,c\}\} & \{\{a,b\}, \{a,e\}, \{b,d\}, \{d,e\}\} \\ \{\{c,d\}\} & \{\{a,c,d\}\} & \{\{b,c,d\}\} & \{\{d,e\}, \{d,c\}\} \\ \{\{e\}\} & \{\{a,e\}\} & \{\{b,c,e\}\} & \{\{d,e\}\} \\ \emptyset & \{\{a\}\} & \{\{b,c\}\} & \{\{d,e\}\} \end{pmatrix} \\ A^3 &= \begin{pmatrix} \emptyset & \{\{a\}\} & \{\{b\}\} & \emptyset \\ \emptyset & \emptyset & \{\{c\}\} & \{\{d\}\} \\ \emptyset & \emptyset & \emptyset & \{\{e\}\} \\ \emptyset & \emptyset & \emptyset & \emptyset \end{pmatrix} \end{aligned}$$

Im Folgenden sei $B = A^*$. Die Matrix B ist, wie bereits gezeigt, die Lösung eines linea-

ren Gleichungssystems. B kann somit mit Hilfe des gaußschen Eliminationsverfahren bestimmt werden. Bei Anwendung dieses Verfahrens auf die Gleichung (3.11) muss in jedem Iterationsschritt folgende Gleichung berechnet werden:

$$B_{ij} = B_{ij} \otimes (B_{ik} \oplus B_{kj})$$

Da nur die s,v-Schnitte, für einen festen Knoten s , von Interesse sind, vereinfacht sich die Gleichung zu:

$$B_j = B_j \otimes (B_k \oplus a_{kj})$$

Im Algorithmus 3 sind die einzelnen Schritte zur Berechnung der minimalen s,v-Schnitte aufgeführt. Nach dem Beenden des Algorithmus sind in der Menge B_t alle minimalen s,t-Schnitte im Graphen zu finden. Dabei wird mit $\Gamma^-(i)$ die Menge aller aus dem Knoten i auslaufenden Bögen bezeichnet.

Algorithmus 3 generate_st_cuts(G, s)

Eingabe: Graph $G=(V,E)$, Startknoten s , Endknoten t

```

for  $i \in V - \{s\}$  do
     $B_i = \{\emptyset\}$ 
end for
 $L = [s]$ 
while  $L \neq []$  do
    Entferne  $i$  aus  $L$ 
    for  $(i, j) \in \Gamma^-(i)$  do
         $T := B_j \otimes (B_i \oplus a_{ij})$ 
        if  $T \neq B_j$  then
             $B_j = T$ 
            Füge  $j$  zu  $L$  hinzu
        end if
    end for
end while
Ausgabe:  $B$ 

```

Beispiel 3.15 Die einzelnen Schritte des Algorithmus sollen wieder am Beispiel des Graphen aus Abbildung 3.12 verdeutlicht werden.

Iterationsschritt	Operation	L
Init	$B = \{\emptyset, \{\emptyset\}, \{\emptyset\}, \{\emptyset\}\}$	$L=[1]$
1	$B(2) = \{\emptyset\} \oplus (\emptyset \otimes \{a\}) = \{a\}$ $B(3) = \{\emptyset\} \oplus (\emptyset \otimes \{b\}) = \{b\}$	$L=[2,3]$
2	$B(3) = \{b\} \oplus (\{a\} \otimes \{c\}) = \{\{a,b\}, \{b,c\}\}$ $B(4) = \{\emptyset\} \oplus (\{a\} \otimes \{d\}) = \{\{a\}, \{d\}\}$	$L=[3,4]$
3	$B(4) = \{a,d\} \oplus (\{\{a,b\}, \{b,c\}\} \otimes \{e\})$ $= \{\{a,b\}, \{a,e\}, \{b,c,d\}, \{d,e\}\}$	$L=[4]$

Tabelle 3.1: Berechnung der Schnitte im Standardgraphen

Als Ergebnis liefert der Algorithmus

$$B = [\emptyset, \{\{a\}\}, \{\{a,b\}, \{b,c\}\}, \{\{a,b\}, \{a,e\}, \{b,c,d\}, \{d,e\}\}]$$

Somit gibt es vier 1,4-Schnitte im Graphen: $B(4) = \{\{a,b\}, \{a,e\}, \{b,c,d\}, \{d,e\}\}$.

3.4 Zuverlässigkeit in speziellen Graphenklassen

In diesem Abschnitt wird das s,t-Zuverlässigkeitsproblem für spezielle Graphen betrachtet. Aufgrund der Komplexität des Zuverlässigkeitsproblems ist nicht davon auszugehen, effiziente Algorithmen für allgemeine Graphen zu finden. In speziellen Graphenklassen kann das Zuverlässigkeitsproblem aber effizient gelöst oder die spezielle Struktur für Vereinfachung verwendet werden. Einige dieser Graphen sind: Der vollständige Graph 3.4.1, der Fächergraph 3.4.3, der Fibonaccigraph 3.4.6 und azyklische Graphen 3.4.9.

3.4.1 Vollständige Graphen

Für den gerichteten, vollständigen Graphen K_n lässt sich eine Rekurrenzgleichung zur Berechnung der s,t-Zuverlässigkeit herleiten. Gerichtete vollständige Graphen besitzen von jedem Knoten aus einen Bogen zu jedem anderen Knoten des Graphen. Ein solcher Graph mit fünf Knoten ist in Abbildung 3.16 zu sehen.

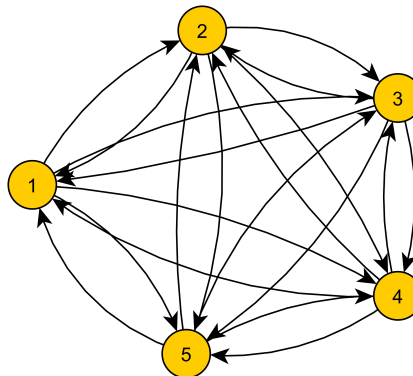


Abbildung 3.16: Vollständiger Graph mit fünf Knoten

Es seien s der Startknoten und t der Zielknoten. Eine Teilmenge der Knoten heißt s-Komponente, wenn alle Knoten dieser Teilmenge vom Knoten s aus erreichbar sind. In einem funktionierenden Zustand muss der Knoten t in der s-Komponente liegen. Eine solche s-Komponente enthält somit zwei bis n Knoten. Also gibt es $\binom{n-2}{j-2}$ Möglichkeiten, die $j-2$ zusätzlichen Knoten in einer Komponente der Größe j zu wählen. Für

jeden dieser j Knoten in der Komponente muss jeweils der aus der Komponente hinauslaufende Bogen zu den $n - j$ Knoten ausgefallen sein. Wäre nämlich ein hinausführender Knoten intakt, würde der Zielknoten auch von s aus erreichbar sein und somit zur s -Komponente gehören. Somit gilt für die s,t-Zusammenhangswahrscheinlichkeit:

$$R_2(K_n) = \sum_{j=2}^n \binom{n-2}{j-2} R_{A_s}(K_j) q^{j(n-j)} \quad (3.12)$$

In dieser Formel kommt die Erreichbarkeit in vollständigen Graphen $R_{A_s}(K_n)$ vor. Sie gibt die Wahrscheinlichkeit an, dass alle n Knoten in einer s -Komponente sind. Um diese zu berechnen, werden die Untergraphen des K_n nach der Anzahl der Knoten aufgeteilt, die in derselben s -Komponente liegen. Für eine feste Komponentengröße j gibt es $\binom{n-1}{j-1}$ Möglichkeiten, Knoten zu wählen, die in einer s -Komponente liegen. Außerdem darf keiner der Bögen, die aus der Komponente hinausführen, intakt sein. Dies führt zu folgender Formel:

$$1 = \sum_{j=1}^n \binom{n-1}{j-1} R_{A_s}(K_j) q^{j(n-j)} \quad (3.13)$$

Umstellen der Formel (3.13) nach $R_{A_s}(K_n)$ führt zu der Formel für die Erreichbarkeit in einem vollständigen Graphen.

$$R_{A_s}(K_n) = 1 - \sum_{j=1}^{n-1} \binom{n-1}{j-1} R_{A_s}(K_j) q^{j(n-j)}$$

Mit Hilfe der Formel (3.12) kann das Zuverlässigkeitspolynom für die s,t-Zusammenhangswahrscheinlichkeit in gerichteten vollständigen Graphen berechnet werden. Der Tabelle 3.2 sind die Polynome für einige vollständige Graphen zu entnehmen.

Knoten	Zuverlässigkeitspolynom
3	$-p^3 + p^2 + p$
4	$-2p^6 + 7p^5 - 7p^4 + 2p^2 + p$
5	$6p^{10} - 42p^9 + 120p^8 - 175p^7 + 127p^6 - 27p^5 - 15p^4 + 3p^3 + 3p^2 + p$
6	$24p^{15} - 276p^{14} + 1422p^{13} - 4310p^{12} + 8464p^{11} - 11132p^{10} + 9699p^9 - 5195p^8 + 1276p^7 + 160p^6 - 126p^5 - 18p^4 + 8p^3 + 4p^2 + p$
7	$-120p^{21} + 2040p^{20} - 16170p^{19} + 79230p^{18} - 268130p^{17} + 662910p^{16} - 1234120p^{15} + 1755230p^{14} - 1911020p^{13} + 1575190p^{12} - 953711p^{11} + 395681p^{10} - 92305p^9 + 815p^8 + 5130p^7 - 370p^6 - 290p^5 - 10p^4 + 15p^3 + 5p^2 + p$

Tabelle 3.2: Zuverlässigkeitspolynome von vollständigen Graphen

3.4.2 Vollständige bipartite Graphen

Auch für vollständige bipartite Graphen lässt sich eine rekursive Formel herleiten. Dazu wird zunächst definiert, was unter einem bipartiten Graphen zu verstehen ist.

Definition 3.16 Ein Graph $G = (V, E)$ heißt bipartit, falls die Knotenmenge V so in zwei Teilmengen V_1 und V_2 zerlegt werden kann, dass für jeden Bogen $(u, v) \in E$ entweder $u \in V_1$ und $v \in V_2$ oder $u \in V_2$ und $v \in V_1$ gilt.

Ein vollständiger bipartiter Graph mit sechs Knoten ist in Abbildung 3.17 zu sehen.

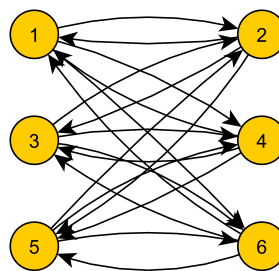


Abbildung 3.17: Vollständiger bipartiter Graph mit sechs Knoten

In einem solchen vollständigen bipartiten Graphen soll jetzt die s,t-Zuverlässigkeit berechnet werden. Der Knoten s ist dabei ein beliebiger Knoten aus der Knotenteilmenge V_1 und der Knoten t ein beliebiger Knoten der Knotenteilmenge V_2 . Weiterhin gilt: $|V_1| = n$ und $|V_2| = m$. Zur Herleitung einer Formel für die s,t-Zuverlässigkeit werden wieder s-Komponenten betrachtet. Da der Knoten t , in einem funktionierenden Zustand, in der s-Komponente liegen muss, kann die s-Komponente zwei bis $n + m$ Knoten enthalten. In der Formel (3.14) gibt j die Größe der s-Komponente und i die Anzahl der Knoten, die zusätzlich zu s in der Knotenteilmenge V_1 ausgewählt werden, an. Um die i zusätzlichen Knoten in V_1 zu wählen gibt es somit $\binom{n-1}{i}$ Möglichkeiten. Die restlichen $j - i - 2$ Knoten der s-Komponente werden aus der Knotenteilmenge V_2 gewählt. Hierfür gibt es $\binom{m-1}{j-i-2}$ Möglichkeiten. Von jedem Knoten aus V_1 gehen $m - (j - i - 2) - 1 = m - j + i + 1$ Bögen zu nicht ausgewählten Knoten aus V_2 und von jedem Knoten aus V_2 gehen $n - i - 1$ Bögen zu nicht ausgewählten Knoten aus V_1 . Diese Bögen gehen aus der s-Komponente hinaus und müssen deshalb ausgefallen sein. Zur Berechnung der s,t-Zuverlässigkeit wird außerdem noch die Wahrscheinlichkeit benötigt, dass diese ausgewählten Knoten eine s-Komponente bilden, das heißt, dass sie alle von s aus erreicht werden können. Diese Wahrscheinlichkeit wird mit $R_{A_s}(K_{i+1, j-i-1})$ bezeichnet. Diese Überlegungen führen auf folgende Formel für die s,t-Zuverlässigkeit in einem vollständigen bipartiten Graphen:

$$R(K_{n,m}) = \sum_{j=2}^{n+m} \sum_{i=0}^{j-2} \binom{n-1}{i} \binom{m-1}{j-i-2} R_{A_s}(K_{i+1,j-i-1}) q^{(i+1)(m-j+i+1)+(j-i-1)(n-i-1)} \quad (3.14)$$

Um die Erreichbarkeit in einem vollständigen bipartiten Graphen zu berechnen, werden die Erreichbarkeiten aller möglichen Untergraphen betrachtet. In der ersten Knotenteilmenge V_1 können, zusätzlich zum Knoten s , 0 bis $n-1$ Knoten ausgewählt werden. Aus der zweiten Knotenteilmenge V_2 können 0 bis m Knoten ausgewählt werden. Sei jetzt j die Anzahl der ausgewählten Knoten aus V_1 und k die Anzahl der Knoten aus V_2 . Es gibt $\binom{n-1}{j-1}$ Möglichkeiten die zusätzlichen Knoten in V_1 zu wählen. Analog gibt es $\binom{m}{k}$ Möglichkeiten die k Knoten in der Knotenteilmenge V_2 auszuwählen. Zusätzlich müssen alle Bögen, die von einem der j ausgewählten Knoten aus V_1 ausgehen, ausgefallen sein. Dies sind genau $i(m-k)$ Bögen. Analog müssen alle $k(n-j)$ Bögen, die ausgewählte Knoten aus V_2 verlassen, ausgefallen sein. Diese Überlegungen führen auf folgende Formel:

$$1 = \sum_{j=1}^n \binom{n-1}{j-1} \sum_{k=0}^m \binom{m}{k} R_{A_s}(K_{j,k}) q^{j(m-k)+k(n-j)} \quad (3.15)$$

Durch umstellen der Formel (3.15) nach $R_{A_s}(K_{n,m})$ ergibt sich folgende Formel für die Erreichbarkeit in einem bipartiten Graphen:

$$R_{A_s}(K_{n,m}) = 1 - \sum_{j=1}^{n-1} \binom{n-1}{j-1} \sum_{k=0}^m \binom{m}{k} R_{A_s}(K_{j,k}) q^{j(m-k)+k(n-j)} - \sum_{k=0}^{m-1} \binom{m}{k} R_{A_s}(K_{n,k}) q^{j(m-k)}$$

Mit Hilfe der Formel (3.14) wurden für einige ausgewählte bipartite Graphen die s,t-Zuverlässigkeiten berechnet. Die so berechneten Zuverlässigkeitspolynome sind in Tabelle 3.3 zu finden.

3.4.3 Fächergraphen

Eine sehr spezielle Klasse von Graphen sind die Fächergraphen F_n . In Abbildung 3.18 ist ein solcher Graph mit n Knoten zu sehen. In diesem Graphen soll die 1,n-Zuverlässigkeit $R_{1,n}(F_n)$ berechnet werden. Für die Berechnung können die vorhandenen Symmetrien und der regelmäßige Aufbau des Graphen verwendet werden. Diese Berücksichtigung führt auf eine schöne rekursive Formel für Fächergraphen.

Größe	Zuverlässigkeitspolynom
2,2	$-p^4 + p^3 + p$
2,3	$p^6 - p^5 - 2p^4 + 2p^3 + p$
3,3	$5p^9 - 21p^8 + 30p^7 - 14p^6 - 4p^4 + 4p^3 + p$
3,4	$13p^{12} - 79p^{11} + 186p^{10} - 195p^9 + 49p^8 + 68p^7 - 45p^6 + 3p^5 - 6p^4 + 6p^3 + p$
3,5	$29p^{15} - 225p^{14} + 716p^{13} - 1140p^{12} + 774p^{11} + 228p^{10} - 712p^9 + 322p^8 + 92p^7 - 92p^6 + 8p^5 - 8p^4 + 8p^3 + p$
4,4	$-73p^{16} + 703p^{15} - 2961p^{14} + 7089p^{13} - 10440p^{12} + 9444p^{11} - 4767p^{10} + 888p^9 + 75p^8 + 168p^7 - 144p^6 + 18p^5 - 9p^4 + 9p^3 + p$

Tabelle 3.3: Zuverlässigkeitspolynome von vollständig bipartiten Graphen

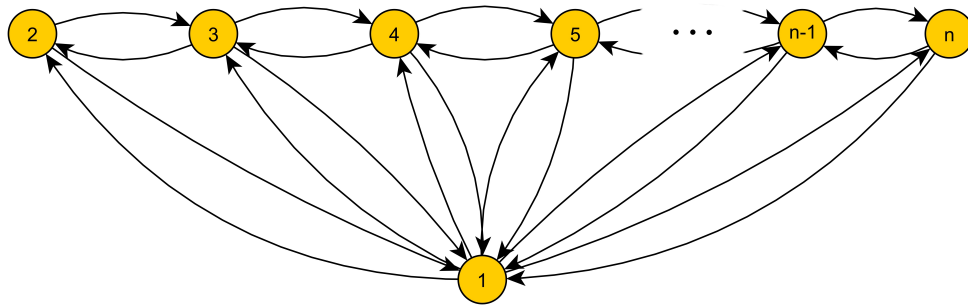


Abbildung 3.18: Fächergraph

Zunächst wird eine Rekurrenzgleichung für das s,t-Zuverlässigkeitspolynom hergeleitet. Der Knoten n im Fächergraphen kann vom Knoten 1 entweder über den Bogen $(1, n)$ oder den Bogen $(n-1, n)$ erreicht werden. Der Bogen $(1, n)$ stellt eine direkte Verbindung dar und ist mit der Wahrscheinlichkeit $p_{(1,n)}$ verfügbar. Wenn der Knoten n über den Bogen $(n-1, n)$ erreicht werden soll, muss zunächst der Knoten $n-1$ erreicht werden. Dies geschieht mit der Wahrscheinlichkeit $R_{1,n-1}(F_{n-1})$. Da die beiden Wege über die Bögen $(1, n)$ und $(n-1, n)$ disjunkt sind, kann, mit Hilfe der Formel für die Reduktion zweier paralleler Bögen aus Abbildung 3.1, die folgende Rekurrenzgleichung formuliert werden. Dabei wird angenommen, dass alle Bögen mit der identischen Wahrscheinlichkeit p verfügbar sind.

$$\begin{aligned}
 R(F_n) &= p + pR(F_{n-1}) - p^2R(F_{n-1}) \\
 &= p + R(F_{n-1})(p - p^2)
 \end{aligned} \tag{3.16}$$

Der Anfangswert für diese Rekurrenzgleichung ist mit $R_{1,2}(F_2) = p$ gegeben. Mit Hilfe der erzeugenden Funktionen kann die Rekurrenzgleichung gelöst werden und erhält folgende explizite Formel für die 1,n-Zuverlässigkeit in einem Fächergraphen mit n Knoten:

$$R(F_n) = \frac{-p + p^2 + ((1-p)p)^n}{p^3 - 2p^2 + 2p - 1} \tag{3.17}$$

Soll die 2,1-Zuverlässigkeit berechnet werden, führt dies wieder auf die Gleichung (3.16) und somit auf die soeben gezeigte explizite Formel.

Wird angenommen, dass der Zielknoten nicht am äußeren Rand des Fächergraphen ist, kann trotzdem eine rekursive Formel hergeleitet werden. Im Fächergraphen in Abbildung 3.19 soll die 1,k-Zuverlässigkeit berechnet werden.

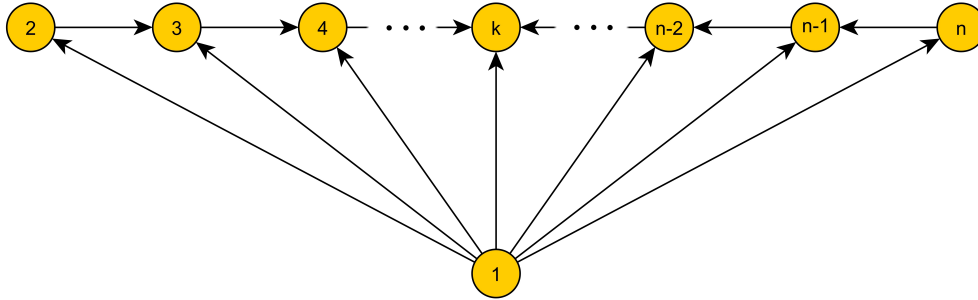


Abbildung 3.19: Fächergraph

Der Knoten k kann über den Bogen $(1, k)$, den Bogen $(k-1, k)$ und den Bogen $(k+1, k)$ erreicht werden. Im ersten Fall ist der Bogen eine direkte Verbindung. Im zweiten Fall muss zunächst der Knoten $k-1$ erreicht werden. Dies geschieht mit der Wahrscheinlichkeit $R(F_{k-1})$ des normalen Fächergraphen. Analog dazu ist der dritte Fall. Auch hier wird der Knoten $n+1$, der zunächst erreicht werden muss, mit der Wahrscheinlichkeit $R(F_{n-k+1})$ erreicht. Diese drei disjunkten Fälle können zu einer Formel zusammengefasst werden, die nur von den Wahrscheinlichkeiten des vorhin behandelten Fächers abhängt.

$$\begin{aligned} R(F_n^k) &= pR(F_{k-1}) + pR(F_{n-k+1}) - p^2R(F_{k-1})R(F_{n-k+1}) + p \\ &\quad - p(pR(F_{k-1}) + pR(F_{n-k+1}) - p^2R(F_{k-1})R(F_{n-k+1})) \\ &= p + R(F_{k-1})(p - p^2) + R(F_{n-k+1})(p - p^2) + R(F_{k-1})R(F_{n-k+1})(p^3 - p^2) \end{aligned}$$

Das Einsetzen der expliziten Lösungsformel (3.17) für den normalen Fächergraphen führt zu folgender Formel.

$$\begin{aligned} R(F_n^k) &= - \frac{p}{(1-p+p^2)^2(p-p^2)^{k+1}} [(1-p)^{3+n}p^{2+n} + (1-p)((1-p)p)^{2k} \\ &\quad + (1-p)^k p^{1+k} (-1 + p(1 + (-1+p)^2 p + ((1-p)p)^n))] \end{aligned}$$

Sind Start- und Zielknoten beide auf dem Rand des Fächergraphen, soll also die 2,n-Zuverlässigkeit berechnet werden, so sind noch einige Überlegungen notwendig. Der Knoten 2 kann entweder über den Bogen $(2, 1)$ oder den Bogen $(2, 3)$ verlassen werden. Ist der Bogen $(2, 1)$ nicht intakt, berechnet sich die Wahrscheinlichkeit des Zustandes in Abhängigkeit von der 2,n-Zuverlässigkeit in einem Fächergraphen mit $n-1$

Knoten mit $(1 - p_{(2,1)})p_{(2,3)}R_{2,n}(F_{n-1})$. Ist der Bogen $(2, 1)$ intakt, kann er kontrahiert werden. Dadurch entstehen zwei parallele Bögen vom Knoten 1 zum Knoten 3. In diesem Fächergraphen mit $n - 1$ Knoten ist jetzt die 1,n-Zuverlässigkeit gesucht. Aufgrund der zwei parallelen Bögen kann aber nicht die bereits hergeleitete explizite Formel angewandt werden. Die Rekurrenzgleichung (3.16) muss erneut mit dem Anfangswert $R_{1,2}(F_2) = 2p - p^2$ gelöst werden. Diese Überlegungen führen zu folgender Formel für die 2,n-Zuverlässigkeit:

$$\begin{aligned} R_{2,n}(F_n) &= (1 - p)pR_{2,n}(F_{n-1}) + p \frac{p^2 + ((1 - p)p)^{n-1} - p((1 - p)p)^{n-1}}{p^3 - p^2 + p} \\ &= (p - p^2)R_{2,n}(F_{n-1}) + \frac{p^2 + ((1 - p)p)^{n-1} - p((1 - p)p)^{n-1}}{p^2 - p + 1} \end{aligned}$$

Diese Rekurrenzgleichung kann mit dem Anfangswert $R_{1,2}(F'_3) = p + p^2 - p^3$ gelöst werden und führt zu folgender expliziter Formel:

$$R_{2,n}(F_n) = \frac{p^4 + (p - p^2)^n(1 + p(n - 2 - (n - 3)p + (n - 2)p^2))}{p^2(p^2 - p + 1)^2} \quad (3.18)$$

3.4.4 Radgraphen

Eine weitere spezielle Klasse von Graphen sind die Radgraphen. Sie entstehen aus den Fächergraphen, indem zwischen dem Knoten 2 und n ein antiparalleles Bogenpaar und zu allen äußeren Bögen jeweils ein antiparalleler Bogen hinzugefügt wird. Ein solcher Radgraph R_8 mit acht Knoten ist in Abbildung 3.20 zu sehen.

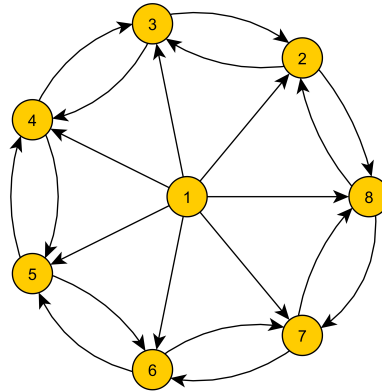


Abbildung 3.20: Radgraph mit acht Knoten

Zur Berechnung der 1,n-Zuverlässigkeit wird die Anzahl der intakten Speichen des Rades betrachtet. Ist der Bogen $(1, n)$ intakt, existiert ein Weg, die Wahrscheinlichkeit hierfür ist p . In diesem Fall interessiert der Zustand der anderen Bögen des Graphen nicht. Jetzt sei also eine andere Speiche als $(1, n)$ intakt und alle anderen Speichen seien

ausgefallen. Dies geschieht mit der Wahrscheinlichkeit $p(1-p)^{n-2}$. Von dem jetzt erreichten Knoten gibt es zwei disjunkte Wege zum Knoten n . Die äußeren Bögen des Rades können im Uhrzeigersinn und gegen den Uhrzeigersinn abgelaufen werden. Sei nun i der durch die intakte Speiche erreichte Knoten. Somit müssen entweder $i-1$ Bögen im Uhrzeigersinn intakt sein, oder $n-i$ Bögen gegen den Uhrzeigersinn. Diese Überlegungen führen zu folgender Formel für das Ereignis, dass eine Speiche intakt ist und es einen intakten 1,n-Weg gibt.

$$\begin{aligned}
 P(E_1) &= p(1 + (1-p)^{n-2} \sum_{i=2}^{n-1} (p^{i-1} + p^{n-i} - p^{n-1})) \\
 &= p \left(1 + (1-p)^{n-1} \left(\frac{-2p^2 + p^n(n+2p-np)}{(-1+p)p} \right) \right) \\
 &= p \left(1 + \frac{(1-p)^{n-3}(-2p^2 + p^n(n+2p-np))}{p} \right) \quad (3.19)
 \end{aligned}$$

Sind zwei oder mehr Speichen intakt, interessieren nur die beiden erreichten Knoten i und j die dem Zielknoten n am nächsten liegen, da es in der Berechnung der Wahrscheinlichkeit nur auf diese ankommt. Es müssen somit alle $i-2$ Speichen zwischen i und n , alle $n-j-1$ Speichen zwischen j und n und die Speiche $(1,n)$ ausgefallen sein. Der Zustand der anderen Speichen ist für die Berechnung nicht von Bedeutung, da es nur auf die beiden nächsten erreichten Knoten ankommt. Von diesen beiden Knoten i und j aus wird jetzt die Wahrscheinlichkeit berechnet, dass ein intakter Weg zum Knoten n existiert. Es müssen vom Knoten i aus $i-1$ Bögen und vom Knoten j aus $n-j$ Bögen intakt sein. Diese Überlegungen führen zu folgender Formel:

$$P(E_2) = p^2 \left(\sum_{i=2}^{n-2} \sum_{j=i+1}^{n-1} (p^{i-1} + p^{n-j} - p^{n-j+i-1})(1-p)^{n+i-j-2} \right) \quad (3.20)$$

Die Summe der beiden soeben ermittelten Gleichungen (3.19) und (3.20) liefert die gesuchte Lösung für die 1,n-Zuverlässigkeit in einem Radgraphen.

$$\begin{aligned}
 R_{1,n}(R_n) &= p \left(1 + \frac{(1-p)^{n-3}(-2p^2 + p^n(n+2p-np))}{p} \right) \\
 &\quad + p^2 \left(\sum_{i=2}^{n-2} \sum_{j=i+1}^{n-1} (p^{i-1} + p^{n-j} - p^{n-j+i-1})(1-p)^{n+i-j-2} \right) \\
 &= \frac{(p^2 - p)(p^3 - p^2 + 1) + (p - p^2)^n((1-p)(p^2 - p + 1)n + (2p^2 - 3p + 2)p)}{(p-1)(p^2 - p + 1)^2}
 \end{aligned}$$

Jetzt soll der Fall betrachtet werden, dass der Startknoten und der Zielknoten adjazente, äußere Knoten im Radgraphen sind. Es soll also die 2,3-Zuverlässigkeit im Graphen der Abbildung 3.20 berechnet werden. Der Knoten 3 kann vom Knoten 2 entweder direkt über den Bogen $(2,3)$ oder über einen Weg im entstehenden Fächergraphen, wenn der

Bogen $(2, 3)$ ausgefallen ist, erreicht werden. Für die Berechnung kann somit die schon hergeleitete Formel (3.18) für einen Fächergraphen mit n Knoten verwendet werden. Die Wahrscheinlichkeit berechnet sich somit wie folgt:

$$\begin{aligned} R_{2,3}(R_n) &= p + R_{2,3}(F_n) - pR_{2,3}(F_n) \\ &= \frac{1}{(p^2(p^2 - p + 1)^2)} (p^3(1 + (p^2 - p)(p^2 - p + 1)) \\ &\quad + (1 - p)(p - p^2)^n(1 + p(-2 + n + (3 - n)p + (-2 + n)p^2))) \end{aligned}$$

3.4.5 Gittergraphen

Gittergraphen G_n sind einfach zu handhabende Graphen, für die rekursive Formeln zur Berechnung der s,t-Zuverlässigkeit hergeleitet werden können. Bei der Herleitung der Formeln kommt es dabei auf die Lage der Start- und Zielknoten im Gittergraphen an. In Abbildung 3.21 ist ein Gittergraph zu sehen, bei dem sich sowohl der Start- als auch der Zielknoten am Anfang des Gittergraphen befinden. Ein Gittergraph G_n hat $2n$ Knoten und $n + 2(n - 1) = 3n - 2$ Bögen.

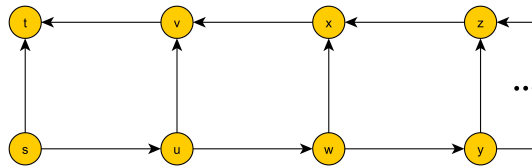


Abbildung 3.21: Gittergraph

Für diese Anordnung der Knoten s und t kann analog zum Vorgehen aus Abschnitt 3.4.3 eine Rekurrenzgleichung hergeleitet werden. Der Knoten t kann von s aus entweder direkt über den Bogen (s, t) oder über die beiden Knoten u und v erreicht werden. Im zweiten Fall müssen die beiden Bögen (s, u) und (v, t) intakt sein. Die Wahrscheinlichkeit, dass u und v miteinander verbunden sind, wird im restlichen Gittergraphen berechnet. Besteht der Gittergraph nur noch aus zwei Knoten, sind diese mit der Wahrscheinlichkeit p verbunden. Diese Überlegungen führen zu folgender Formel für die s,t-Zuverlässigkeit:

$$\begin{aligned} R(G_n) &= p + p^2 R(G_{n-1}) - p^3 R(G_{n-1}) \\ &= p + R(G_{n-1})(p^2 - p^3) \end{aligned}$$

Auch hier kann wieder mit Hilfe von erzeugenden Funktionen eine explizite Formel angegeben werden:

$$R(G_n) = \frac{p - p((1 - p)p^2)^n}{p^3 - p^2 + 1} \quad (3.21)$$

Die zweite Möglichkeit, wie die beiden Knoten s und t angeordnet sein können, ist,

wenn sie im Gittergraphen gegenüberliegen. Dieser Fall ist in Abbildung 3.22 zu sehen. Im Gittergraphen $G_n^{m,l}$ sind n Reihen vorhanden, wobei l links und m rechts von den Knoten s und t liegen.

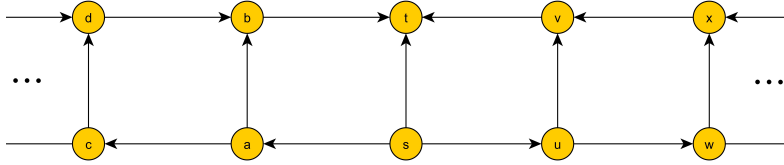


Abbildung 3.22: Gittergraph

In diesen Fall kann man die s,t-Zuverlässigkeit auf den ersten Fall zurückführen. Die beiden Knoten s und t können entweder direkt über den Bogen (s, t) miteinander verbunden sein, oder über die Knoten a und b oder über die Knoten u und v . Im zweiten Fall muss somit der Bogen (s, a) und der Bogen (b, t) intakt sein. Die Wahrscheinlichkeit, dass a und b verbunden sind, berechnet sich über die Gleichung (3.21) für einen Gittergraphen der Länge l . Die Berechnung des Weges über die Knoten u und v funktioniert analog. Diese Überlegungen führen zu folgender Gleichung. Durch Einsetzen der bereits bekannten Formel (3.21) und Zusammenfassen erhält man eine explizite Formel für diese Start- und Endknotenanzahl im Gittergraphen.

$$\begin{aligned}
 R(G_n^{m,l}) &= (p + p^2 R(G_l) - p^3 R(G_l)) + p^2 R(G_m) - p^2 R(G_m) (p + p^2 R(G_l) - p^3 R(G_l)) \\
 &= \frac{1}{(1 - p^2 - p^3)^2} p (1 - (p^2 - p^3) \\
 &\quad (p^2 + (p^2 - p^3)^l - p^2 (p^2 - p^3)^l + (p^2 - p^3)^m (1 + (-p^2 + p^3 (p^2 - p^3)^l))))
 \end{aligned}$$

3.4.6 Fibonaccigraphen

Ein Fibonaccigraph ist ein gerichteter Graph $G = (V, E)$ mit n Knoten, bei dem jeder Knoten v , bis auf die Knoten $n-1$ und n , zwei auslaufende Bögen $(v, v+1)$ und $(v, v+2)$ besitzt. Ein solcher Graph ist in Abbildung 3.23 zu sehen. Für Fibonaccigraphen haben Korenblit und Levit [Kor02] gezeigt, dass mit Hilfe des in Abschnitt 3.3.4 vorgestellten Algorithmus, das s,t-Zuverlässigkeitsproblem in polynomialer Zeit lösbar ist.

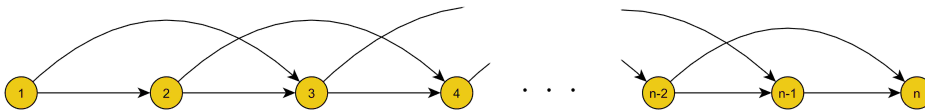


Abbildung 3.23: Fibonaccigraph

Im Folgenden wird ein Satz über die Anzahl der Minimalschnitte in einem Fibonaccigraphen bewiesen. Dazu sind noch einige Vorbetrachtungen nötig. Sei $FG(i, j)$, mit $i, j \in V$ und $i < j$, der Untergraph des Fibonaccigraphen FG vom Knoten i bis zum Knoten j . Somit ist also der $FG(i, j-1)$ ein Untergraph von $FG(i, j)$. Ein Schnitt im Graphen

$FG(i, j)$, der auch den Zusammenhang des Untergraphen $FG(i, j - 1)$ zerstört, heißt *strenger Minimalschnitt*. Alle Schnitte, die keinen Einfluss auf den Zusammenhang des Untergraphen haben, heißen *schwache Minimalschnitte*. Die Menge der Minimalschnitte des Graphen $FG(i, j)$ wird mit $CF(i, j)$, die Menge der *starken Minimalschnitte* mit $CF(i, j - 1, j)$ und die Menge der *schwachen Minimalschnitte* mit $CF(i, \overline{j - 1}, j)$ bezeichnet.

Für einen Fibonaccigraphen mit n Knoten gilt somit folgende Gleichung:

$$CF(1, n) = CF(1, n - 1, n) \cup CF(1, \overline{n - 1}, n) \quad (3.22)$$

Mit Hilfe dieser Vorbetrachtungen können nun die folgenden Lemmata formuliert werden. Die Beweise sind in [Kor02] zu finden.

Lemma 3.17 $|CF(1, n - 1, n)| = |CF(1, n - 1)|$, für $n \geq 3$.

Lemma 3.18 $|CF(1, \overline{n - 1}, n)| = \lfloor \frac{n}{2} \rfloor$, für $n \geq 2$.

Mit Hilfe der beiden Lemmata kann nun der folgende Satz bewiesen werden:

Satz 3.19 Für $n \geq 2$ gilt für die Anzahl der Minimalschnitte in einem Fibonaccigraphen mit n Knoten: $\mu = \lfloor \frac{n^2}{4} \rfloor$.

Beweis: Für $n = 2$ und $n = 3$ ist die Behauptung klar. Im Folgenden sei $n > 3$. Somit gilt nach Gleichung (3.22) und den Lemmata 3.17 und 3.18:

$$\begin{aligned} |CF(1, n)| &= |CF(1, n - 1, n)| + |CF(1, \overline{n - 1}, n)| \\ &= |CF(1, n - 1)| + \lfloor \frac{n}{2} \rfloor \\ &= |CF(1, n - 1 - 1)| + \left\lfloor \frac{n - 1}{2} \right\rfloor + \lfloor \frac{n}{2} \rfloor \\ &= |CF(1, n - 2)| + n - 1 \end{aligned}$$

Für gerade n können, durch wiederholtes Anwenden der soeben gezeigten Beziehung

$|CF(1, n)| = |CF(1, n-2)| + n-1$, folgende Umformungsschritte ausgeführt werden:

$$\begin{aligned}
 |CF(1, n)| &= |CF(1, n-2)| + n-1 \\
 &= |CF(1, n-2-2)| + n-2-1 + n-1 \\
 &= |CF(1, n-4)| + 2n-4 \\
 &= |CF(1, n-6)| + 3n-9 \\
 &= \dots \\
 &= |CF(1, n-(n-2))| + \frac{n-2}{2}n - \left(\frac{n-2}{2}\right)^2 \\
 &= |CF(1, 2)| + \frac{n^2-4}{4} \\
 &= 1 + \frac{n^2}{4} - 1 = \frac{n^2}{4} = \left\lfloor \frac{n^2}{4} \right\rfloor
 \end{aligned}$$

Mit Hilfe dieses Ergebnisses gilt für ungerade n :

$$\begin{aligned}
 |CF(1, n)| &= |CF(1, n-1)| + \left\lfloor \frac{n}{2} \right\rfloor \\
 &= \frac{(n-1)^2}{4} + \frac{n-1}{2} = \frac{n^2-1}{4} = \left\lfloor \frac{n^2}{4} \right\rfloor
 \end{aligned}$$

□

Nach Satz 3.19 ist die Anzahl der Minimalschnitte somit $O(n^2)$. Der in Abschnitt 3.3.4 vorgestellte Algorithmus hat die Komplexität $O((m+n)\mu^2)$. Der Fibonaccigraph hat $m = 2n-3$ Kanten. Somit kann der Aufwand des Schnitt-Algorithmus für Fibonaccigraphen mit $O((m+n)\mu^2) = O((2n-3+n)n^4) = O(n^5)$ abgeschätzt werden.

3.4.7 BSP-Graphen

BSP-Graphen (engl.: basically-series-parallel) sind eine Erweiterung der Serien-Parallelen-Graphen für den gerichteten Fall. Für die s,t-Zuverlässigkeit haben Agrawal und Satyanarayana [Agr84a] einen Algorithmus angegeben, der nur mit Hilfe von Reduktionen die s,t-Zuverlässigkeit berechnet. In BSP-Graphen kann somit die s,t-Zuverlässigkeit, analog zu Serien-Parallelen Graphen, sehr einfach berechnet werden. Die folgende Definition gibt an, wann ein Graph ein BSP-Graph ist.

Definition 3.20 Ein gerichteter Graph heißt BSP-Graph, falls der zugrunde liegende ungerichtete Graph ein Serien-Paralleler Graph ist. Ein Serien-Paralleler Graph $G = (V, E)$ ist ein ungerichteter Graph, der durch wiederholtes Anwenden der Serien- und Parallelreduktion zwischen zwei Knoten s und t des Graphen, in die Kante (s, t) überführt werden kann.

Ein BSP-Graph besitzt einige besondere Eigenschaften, die für die Berechnung der s,t-Zuverlässigkeit verwendet werden können. Die beiden folgenden Sätze beschreiben diese Besonderheiten.

Satz 3.21 *Sei $G = (V, E)$ ein BSP-Graph ohne Artikulation, mit $|V| \geq 3$. Dann enthält G mindestens zwei Knoten mit genau zwei Nachbarn.*

Beweis: Sei G_1 der G zugrunde liegende ungerichtete Graph. Im Graphen G_1 werden alle parallelen Kanten mit Hilfe der Parallelreduktion durch eine einzelne Kante ersetzt. Zwei Knoten in G sind genau dann benachbart, falls sie in G_1 adjazent sind. Somit reicht es aus zu zeigen, dass im Graphen G_1 mindestens zwei Knoten mit Knotengrad 2 existieren. Der Beweis wird mit Hilfe von vollständiger Induktion nach der Anzahl der Knoten in G_1 geführt.

Falls G_1 genau drei Knoten besitzt, hat jeder Knoten den Knotengrad 2. Sei die Behauptung für alle Graphen G_1 mit n Knoten bewiesen. G_1 sei ein Graph mit $n + 1$ Knoten. Der Graph G_1 muss mindestens einen Knoten vom Grad 2 haben, da sonst keine Serien- und keine Parallelreduktionen möglich wären und der Graph somit nicht Serien-Parallel wäre. Angenommen, der Graph G_1 hätte nur einen Knoten u vom Grad 2 und seien v und w die zwei adjazenten Knoten zu u . Auf den Knoten u wird jetzt eine Serienreduktion angewendet und eventuell entstehende parallele Kanten durch eine Parallelreduktion entfernt. Der so entstehende Graph sei der Graph G_2 . Die Knoten v und w sind im Graphen G_2 adjazent und der Graph G_2 besitzt n Knoten. Somit existieren nach Induktionsvoraussetzung mindestens zwei Knoten vom Grad 2 in G_2 . Da u der einzige Knoten in G_1 ist, der den Knotengrad 2 hat, müssen in G_2 v und w den Knotengrad 2 haben. Sei v_1 der zweite zu v adjazente Knoten und w_1 der zweite zu w adjazente Knoten. Auf den Knoten v wird jetzt wieder eine Serien-Reduktion angewendet. Der so entstehende Graph sei der Graph G_3 . Der Graph G_3 hat $n - 1$ Knoten, aber nur der Knoten w hat den Knotengrad 2, dies ist ein Widerspruch zur Induktionsannahme. Somit muss in G_2 noch ein anderer Knoten als v und w den Knotengrad 2 haben. Also muss der Graph G_1 mindestens zwei Knoten vom Grad 2 besitzen. \square

Satz 3.22 *Sei $G = (V, E)$ ein BSP-Graph ohne Artikulation, mit $|V| \geq 3$ und $s, t \in V$. Falls G keine Knoten, außer den Knoten s und t , mit zwei Nachbarn besitzt, hat G ein s -Dreieck und mindestens ein Knoten in diesem Dreieck hat genau drei Nachbarn.*

Beweis: G_1 sei der G zugrundeliegende ungerichtete Graph. Im Graphen G_1 werden alle parallelen Kanten mit Hilfe der Parallelreduktion durch eine einzelne Kante ersetzt. Da s und t die einzigen Knoten mit zwei Nachbarn in G sind, sind s und t die einzigen beiden Knoten mit Knotengrad 2 in G_1 . Seien v und w die beiden Knoten, die zu s adjazent sind. Es bleibt zu zeigen, dass v und w adjazent sind und dass mindestens einer der beiden Knoten genau den Knotengrad 2 hat.

Zuerst wird gezeigt, dass v und w adjazent sind. Dazu wird angenommen, dass v und

w nicht adjazent sind. Der Graph G_2 entsteht aus dem Graphen G_1 , indem eine Serien-Reduktion auf den Knoten s durchgeführt wird. Nach der Reduktion sind die Knoten v und w in G_2 adjazent. Der Knotengrad ändert sich bei dieser Operation nicht. Somit ist wie in G_1 , t der einzige Knoten mit Knotengrad 2. Dies ist ein Widerspruch zum Satz 3.21, da mindestens zwei Knoten mit Knotengrad 2 im Graphen sind.

Um die zweite Behauptung zu beweisen, wird angenommen, dass beide Knoten v und w einen Knotengrad größer als 3 haben. Der Graph G_2 wird wieder aus G_1 durch Durchführung einer Serien-Reduktion gewonnen. Da die beiden Knoten v und w adjazent sind, entsteht durch die Serien-Reduktion genau ein paralleles Bogenpaar. G_3 sei der Graph, der aus G_2 entsteht, wenn die beiden parallelen Kanten durch eine einzelne ersetzt werden. Falls in G_1 für die Knotengrade der Knoten v und w gilt: $d_v > 3$ und $d_w > 3$, dann gilt für die Knotengrade im Graphen G_3 : $(d_v - 1) > 2$ und $(d_w - 1) > 2$. Somit hat nur der Knoten t im Graphen G_3 den Grad 2. Dies steht im Widerspruch zum Satz 3.21. Also muss in G_1 mindestens eine der beiden Knoten v und w den Knotengrad 3 haben. \square

Die beiden Besonderheiten der BSP-Graphen, die in den Sätzen 3.21 und 3.22 gezeigt wurden, können für die Reduktion der BSP-Graphen verwendet werden. In einem BSP-Graphen können somit immer Reduktionen von Knoten mit zwei Nachbarn (siehe Abbildung 3.2) und die s-Dreiecks-Reduktion, die in Abschnitt 3.1 vorgestellt wurde, angewandt werden. Für die Verwendung dieser Reduktionen auf einem BSP-Graphen gilt folgender Satz [Agr84a]:

Satz 3.23 *Sei G ein gerichteter Graph. Der Graph G' entsteht aus G durch Anwendung einer Grad-2-Reduktion (siehe Abbildung 3.2) oder durch eine s-Dreiecks-Reduktion. Falls G ein BSP-Graph ist, so ist auch G' ein BSP-Graph.*

Dieser Satz ist die Grundlage für die Berechnung der s,t-Zuverlässigkeit in einem BSP-Graphen. Die Zuverlässigkeit kann durch wiederholtes Anwenden von Reduktionen berechnet werden. Nach jedem Reduktionsschritt werden eventuell auftretende parallele Bögen mit Hilfe der Parallelreduktion in einen einzelnen Bogen umgewandelt. Agrawal und Satyanarayana [Agr84a] haben gezeigt, dass die s,t-Zuverlässigkeit in BSP-Graphen in $O(|E|)$ Schritten berechnet werden kann.

3.4.8 Graphen mit beschränkter Wegweite

Graphen mit beschränkter Weg- oder Baumweite haben in der praktischen Anwendung eine große Bedeutung. Auf ihnen lassen sich viele NP-schwere Probleme in polynomieller Zeit lösen. So sind zum Beispiel das Hamiltonkreisproblem, also die Frage, ob ein Graph einen Hamiltonkreis besitzt, und das Steinerbaumproblem auf Graphen mit beschränkter Baumweite in polynomieller Zeit lösbar. Für die Berechnung der s,t-Zuverlässigkeit kann die Wegzerlegung eines Graphen benutzt werden. Ein Algorithmus für die Bestimmung des Zuverlässigkeitspolynoms in ungerichteten Graphen findet sich

in [Bei12]. In diesem Abschnitt wird ein neuer Algorithmus für das s,t-Zuverlässigkeitsproblem $Conn_2$ in gerichteten Graphen angegeben. Er ist eine Erweiterung des Algorithmus für den ungerichteten Fall auf den gerichteten Fall. Dazu werden die in [Bei12] angegebenen Datenstrukturen erweitert und für gerichtete Graphen angepasst. Um diesen Algorithmus angeben zu können, sind zunächst noch einige Definitionen nötig.

Definition 3.24 Eine Baumzerlegung eines Graphen $G = (V, E)$ ist ein Paar (T, ϕ) . Dabei ist T ein Baum $T = (W, F)$ und ϕ eine Funktion $\phi : W \rightarrow 2^V$ ist, die jeder Knotenteilmenge von G eine Kante von T zuordnet, so dass

1. $\bigcup_{w \in W} \phi(w) = V$
2. $\{u, v\} \in E \Rightarrow \exists w \in W : \{u, v\} \subseteq \phi(w)$
3. Wenn $v \in W$ auf dem Weg zwischen u und w in T ist, dann gilt: $\phi(u) \cap \phi(w) \subseteq \phi(v)$

Falls T ein Weg ist, heißt die Zerlegung Wegzerlegung.

Die Weite einer Wegzerlegung W ist definiert durch: $\max\{|\phi(w)| : w \in W\} - 1$. Die Wegweite $pw(G)$ eines Graphen G ist das Minimum über die Weiten aller Wegzerlegungen von G . Die Wegzerlegung eines Graphen kann als eine geordnete Sequenz $(X_1, X_2, X_3, \dots, X_l)$ von Knotenteilmengen dargestellt werden.

Definition 3.25 Eine Wegzerlegung heißt *schöne Wegzerlegung*, falls folgende zwei Bedingungen erfüllt sind:

1. $|X_0| = |X_l| = 1$
2. $|X_i \triangle X_{i+1}| = 1 \quad \forall i = 1, \dots, l-1$ Wobei $A \triangle B$ die symmetrische Differenz der Mengen A und B bezeichnet.

Eine *schöne Wegzerlegung* kann somit mit einer Sequenz von Knoten dargestellt werden, die jeden Knoten des Graphen zweimal enthält. Eine *schöne Graphenzerlegung* ist eine Sequenz $S = (s_1, s_2, \dots, s_k)$ aus den Knoten und Bögen des Graphen, falls folgende drei Bedingungen erfüllt sind:

1. Das Löschen aller Bögen aus S ergibt eine *schöne Wegzerlegung*
2. Alle Bögen des Graphen G kommen genau einmal in S vor
3. Falls $s_l = (u, v) \in E$, dann gibt es Indizes i, j, p, q mit $i, j < l$ und $p, q > l$, so dass $u = s_i = s_p$ und $v = s_j = s_q$

Die Bedingung 3 stellt sicher, dass vor dem Auftreten eines Bogens der Start- und Endknoten des Bogens schon genau einmal in der Sequenz aufgetreten ist.

In Abbildung 3.24 ist ein Graph mit der Wegweite zwei zu sehen. Eine *schöne Graphen-*

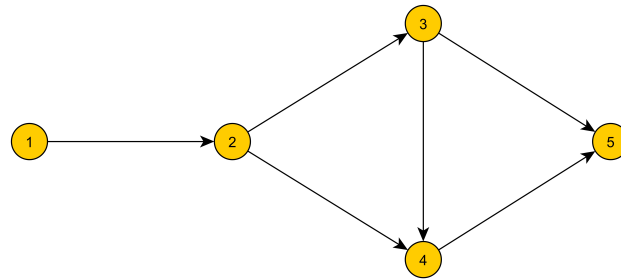


Abbildung 3.24: Graph mit der Wegweite zwei

zerlegung dieses Graphen ist

$$(1, 2, (1, 2), -1, 3, (2, 3), 4, (2, 4), -2, (3, 4), 5, (3, 5), -3, (4, 5), -4, -5)$$

Hier wird jedes zweite Auftreten eines Knotens mit einem Minuszeichen versehen. Die dazugehörige *schöne Wegzerlegung* ist

$$(\emptyset, \{1\}, \{1, 2\}, \{2\}, \{2, 3\}, \{2, 3, 4\}, \{3, 4\}, \{3, 4, 5\}, \{4, 5\}, \{5\}, \emptyset)$$

Diese Knotenteilmengen stellen die aktiven Knoten dar. Für einen Graphen $G = (V, E)$ mit n Knoten lässt sich eine Folge von $2n + 1$ aktiven Knotenteilmengen notieren, bei der die erste und letzte Menge die leere Menge ist. Die Vergrößerung einer aktiven Menge mit einem Knoten v wird die Aktivierung des Knotens v genannt, das Entfernen eines Knotens v die Deaktivierung des Knotens v . Diese Operationen korrespondieren mit dem Auftreten von Knoten mit positiven, beziehungsweise negativen Vorzeichen in S . In S treten außerdem auch Bögen auf, die verarbeitet werden müssen. Die Verarbeitung eines Bogens ändert die aktive Knotenmenge nicht.

Die Idee ist, die Graphenzerlegung zu durchlaufen und schrittweise alle Knoten und Bögen des Graphen zu verarbeiten. In jedem Schritt tritt eine Menge von Zuständen auf. In der folgenden Definition wird angegeben, was unter einem Zustand zu verstehen ist.

Definition 3.26 Ein Zustand Z ist ein Tupel $(G(Z), P(Z))$, wobei G ein Graph und P eine Wahrscheinlichkeit ist. Die Knoten des Graphen G können zusätzlich mit zwei Markierungen, der s -Markierung und der t -Markierung, versehen sein.

Falls ein Knoten in einem Zustand s -markiert ist, bedeutet dies, dass er in diesem Zustand vom Knoten s aus erreicht werden kann. Ist er t -markiert, kann in diesem Zustand von diesem Knoten aus der Knoten t erreicht werden.

Die drei möglichen Aktionen, das Aktivieren und Deaktivieren von Knoten und das Verarbeiten von Bögen, werden immer auf alle Zustände eines Iterationsschrittes angewandt. Dabei können die auftretenden Zustände teilweise zusammengefasst werden,

beziehungsweise sind für die s,t-Zuverlässigkeit nicht relevant und können weggelassen werden. Ohne das Zusammenfassen von Zuständen und ohne dem Weglassen von unnötigen Zuständen führt diese Herangehensweise zur vollständigen Enumeration aller möglichen Zustände. In den Zuständen wird nach dem Verarbeiten eines Bogens überprüft, ob ein nicht markierter Knoten markiert werden kann, das heißt, dass er von einem s-markierten Knoten erreicht werden oder einen t-markierten Knoten erreichen kann. Ist der Algorithmus in der Graphenzerlegung im letzten Element angekommen, enthalten die Graphen der Zustände nur noch einen Knoten. Für die Zuverlässigkeitsberechnung sind jetzt nur die Zustände interessant, in denen der Knoten t von s aus erreicht werden kann, also der einzelne Knoten sowohl s- als auch t-markiert ist. Die s,t-Zuverlässigkeit ist dann die Summe der Wahrscheinlichkeiten dieser Zustände. Der Ablauf des Verfahrens wird im folgenden Algorithmus dargestellt. In diesem Algorithmus wird eine Folge von Mengen $(S_0, S_1, S_2, \dots, S_k)$ zu einer schönen Graphenzerlegung $S = (s_1, s_2, \dots, s_k)$ erzeugt, wobei jedes Element der Menge S_i eine Menge von Zuständen enthält. Die Menge S_0 wird mit dem Zustand $(\emptyset, 1.0)$ initialisiert.

Algorithmus 4 path_decomp_reliability(G, s, t)

Eingabe: Graph $G=(V,E)$, Startknoten s , Endknoten t , schöne Graphenzerlegung S

```

for  $s$  in  $S$  do
  if  $s$  ist ein Bogen then
    verarbeiteBogen( $s$ )
  else if  $s > 0$  then
    aktiviereKnoten( $s$ )
  else
    deaktiviereKnoten( $s$ )
  end if
end for

```

$$R_{st} = \sum_{Z \in S_k} (P(Z))$$

Ausgabe: Zuverlässigkeit $R_{st}(G)$

Im Algorithmus müssen die Funktionen für das Aktivieren und Deaktivieren von Knoten und das Verarbeiten von Bögen definiert werden.

1. **Knoten v aktivieren:** Im Zustand Z wird zum Graphen $G(Z)$ ein neuer Knoten v hinzugefügt.
2. **Knoten v deaktivieren:** Im Zustand Z wird der Knoten v aus dem Graphen $G(Z)$ gelöscht. Befindet sich nach dem Löschen des Knotens v in $G(Z)$ kein s-markierter Knoten mehr, wird der ganze Zustand gelöscht, da dieser und alle nachfolgenden Zustände keinen Beitrag mehr zur s,t-Zuverlässigkeit leisten. Ein Zustand darf aber erst gelöscht werden, wenn der Knoten s schon aktiviert wurde, das heißt, es wurde schon ein Knoten s-markiert. Falls der Knoten t schon aktiviert wurde und keine t-markierten Knoten mehr im Zustand existieren, kann auch hier der ganze Zustand gelöscht werden.

3. **Bogen verarbeiten:** Aus dem Zustand Z werden zwei Zustände Z' und Z'' erzeugt. Im ersten Zustand wird der Bogen nicht eingefügt und die Wahrscheinlichkeit mit $P(Z') = (1 - p_e)P(Z)$ berechnet. Im zweiten Zustand wird der Bogen (v, w) eingefügt und die Wahrscheinlichkeit mit $P(Z'') = p_e P(Z)$ berechnet. Sind beide Endknoten des Bogens gleichartig markiert, so wird kein Bogen eingefügt, da sich die Markierung durch das Einfügen beziehungsweise das Nichteinfügen des Bogens nicht ändert. In diesem Fall ist der neue Zustand gleich dem alten Zustand, also $Z'' = Z' = Z$.

Verschiedene Zustände Z_i und Z_j in einem Iterationsschritt können zusammengefasst werden. Dazu müssen die Knoten, sowie die markierten Knoten der Graphen $G(Z_i)$ und $G(Z_j)$ übereinstimmen. Außerdem müssen alle Bögen, die zwischen unmarkierten Knoten der Graphen $G(Z_i)$ und $G(Z_j)$ verlaufen, übereinstimmen. Stimmen zwei Zustände überein, werden ihre Zustandswahrscheinlichkeiten addiert und einer der beiden Zustände kann gelöscht werden.

Beispiel 3.27 Für den Graphen aus der Abbildung 3.24 soll die s,t-Zuverlässigkeit $R_{1,5}(G)$ berechnet werden. Dazu wird zunächst eine Graphenzerlegung ermittelt. Eine solche Graphenzerlegung ist durch

$$(1, 2, (1, 2), -1, 3, (2, 3), 4, (2, 4), -2, (3, 4), 5, (3, 5), -3, (4, 5), -4, -5)$$

gegeben. Diese Graphenzerlegung führt auf den in Tabelle 3.4 zu sehenden Ablauf des Verfahrens.

Aufwandsanalyse

Die Anzahl der Iterationsschritte in diesem Verfahren hängt von der Länge der Graphenzerlegung ab und ist somit $2n + m$. In jedem Iterationsschritt des Algorithmus entstehen gerichtete, azyklische Graphen. Im schlechtesten Fall müssen alle möglichen azyklischen Graphen mit n Knoten erzeugt werden. Robinson [Rob73] hat 1973 eine rekursive Formel für die Berechnung der Anzahl azyklischer Graphen vorgestellt. In dieser Formel werden Erweiterungen von azyklischen Graphen mit weniger Knoten vorgenommen. Sei ein azyklischer Graph mit $n - k$ Knoten gegeben. Zu diesem Graph werden k Knoten hinzugefügt. Jetzt gibt es $2^{k(n-k)}$ mögliche Bögen zwischen den neuen und den alten Knoten einzufügen. In einem Graphen mit n Knoten gibt es $\binom{n}{k}$ Möglichkeiten, diese k Knoten auszuwählen. Die k Knoten können außerdem markiert oder nicht markiert sein. Mit Hilfe des Inklusion-Exklusion-Prinzips erhält man somit die Formel (3.23) für die Anzahl der azyklischen Graphen mit n Knoten.

$$a_n = \sum_{k=1}^n (-1)^{k-1} \binom{n}{k} 2^{k(n-k)} 2^k a_{n-k} \quad (3.23)$$

n		G(Z)	s-markiert	t-markiert	P(Z)
0		(\emptyset, \emptyset)	\emptyset	\emptyset	1.0
1	+1	$([1], \emptyset)$	$\{1\}$	\emptyset	1.0
2	+2	$([1, 2], \emptyset)$	$\{1\}$	\emptyset	1.0
3	(1, 2)	$([1, 2], \{(1, 2)\})$	$\{1, 2\}$	\emptyset	0.8
		$([1, 2], \emptyset)$	$\{1\}$	\emptyset	0.2
4	-1	$([2], \emptyset)$	$\{2\}$	\emptyset	0.8
5	+3	$([2, 3], \emptyset)$	$\{2\}$	\emptyset	0.8
6	(2, 3)	$([2, 3], \{(2, 3)\})$	$\{2, 3\}$	\emptyset	0.64
		$([2, 3], \emptyset)$	$\{2\}$	\emptyset	0.16
7	+4	$([2, 3, 4], \{(2, 3)\})$	$\{2, 3\}$	\emptyset	0.64
		$([2, 3, 4], \emptyset)$	$\{2\}$	\emptyset	0.16
8	(2, 4)	$([2, 3, 4], \{(2, 3), (2, 4)\})$	$\{2, 3, 4\}$	\emptyset	0.512
		$([2, 3, 4], \{(2, 3)\})$	$\{2, 3\}$	\emptyset	0.128
		$([2, 3, 4], \{(2, 4)\})$	$\{2, 4\}$	\emptyset	0.128
		$([2, 3, 4], \emptyset)$	$\{2\}$	\emptyset	0.032
...
15	-4	$([5], \emptyset)$	$\{5\}$	$\{5\}$	0.713

Tabelle 3.4: Ablauf des Graphenzerlegungsalgorithmus

In Tabelle 3.5 sind für die ersten Werte die Anzahl der azyklischen, markierten Graphen aufgelistet. Sei k die Weite der Wegzerlegung eines Graphen G mit n Knoten und m

n	0	1	2	3	4	5	6	7
a_n	1	1	4	56	2.288	241.952	62.113.728	37.340.491.904

Tabelle 3.5: Anzahl der azyklischen Graphen mit n Knoten

Bögen, dann kann der Aufwand für den Algorithmus 4 mit $O((2n+m)a_k) = O((n+m)a_k)$ abgeschätzt werden. Der Aufwand wächst somit linear in n und m , aber exponentiell in der Weite der Wegzerlegung des Graphen. Das Verfahren ist somit nur auf Graphen mit geringer Wegweite praktikabel anwendbar und es ist von entscheidender Bedeutung für den Algorithmus, eine Wegzerlegung mit geringer Wegweite zu finden.

Eine Heuristik zur Ermittlung einer schönen Graphenzerlegung

Für den Algorithmus 4 wird eine *schöne Graphenzerlegung* benötigt. Um eine solche *schöne Graphenzerlegung* zu erhalten, wird zunächst eine schöne Wegzerlegung ermittelt und anschließend daraus die Graphenzerlegung erzeugt. In der Literatur sind verschiedene Verfahren zur Erzeugung von Wegzerlegungen bekannt. Hisao Tamaki hat in [Tam11] einen Algorithmus vorgestellt, der in $O(mn^{k+1})$ Schritten eine Wegzerlegung mit höchstens der Weite k berechnet. Weitere Verfahren sind in [Cat96], [Bod91],

[Bod98] und [Gog04] zu finden. Die meisten Algorithmen sind darauf ausgelegt zu prüfen, ob ein Graph die Wegweite k hat oder nicht. Im Laufe der Überprüfung erzeugen sie meist auch eine passende Wegzerlegung. Die genannten Algorithmen sind allerdings sehr schwer zu implementieren und sehr komplex. Deshalb wird in diesem Abschnitt eine Heuristik zur Erzeugung einer schönen Graphenzerlegung angegeben.

Der Heuristik liegt folgende Idee zugrunde: Beginnend vom Startknoten s , werden zunächst alle inzidenten Bögen eines Knotens abgearbeitet, bevor ein neuer Knoten in die Graphenzerlegung aufgenommen wird. Sind bei einem Knoten alle Bögen abgearbeitet, wird er in der Graphenzerlegung deaktiviert. Der Ablauf dieses Verfahrens ist im folgenden Algorithmus zu sehen.

Algorithmus 5 `get_graph_decomp(G, s, t)`

Eingabe: Graph $G=(V,E)$, Startknoten s , Endknoten t

$activeNodes = S = [s]$

```

while  $\text{len}(activeNodes) \neq 0$  do
    flag = False                                ▷ Wurde in diesem Schritt ein Bogen hinzugefügt?
    for  $v$  in  $activeNodes$  do                    ▷ Kann ein Knoten deaktiviert werden?
        if  $d^-(v) == 0$  und  $d^+(v) == 0$  then
            deaktiviereKnoten( $v$ )
            entferne  $v$  aus  $activeNodes$ 
        end if
    end for
    for  $v$  in  $activeNodes$  do                    ▷ Suche nach einem Bogen zum Hinzufügen
        for  $e = (v, w)$  do
            if  $w$  in  $activeNodes$  then
                Bogen  $e$  zur Graphenzerlegung  $S$  hinzufügen
                flag = True
            end if
        end for
    end for
    if flag == False then                    ▷ Falls kein Bogen hinzugefügt wurde
        for  $v$  in  $activeNodes$  do
            if  $v$  hat inzidenten Bogen  $(v, w)$ , mit  $w$  nicht in  $activeNodes$  then
                Füge  $w$  zu  $activeNodes$  und zu  $S$  hinzu
            end if
        end for
    end if
end while
Ausgabe: Graphenzerlegung  $S$ 

```

In diesem Algorithmus wird immer mit dem Knoten s als Startknoten der Zerlegung begonnen. Dies kann dazu führen, dass die der Graphenzerlegung zugrunde liegende Wegzerlegung eine große Weite hat, obwohl der Graph an sich eine geringe Wegweite hat. Für den Algorithmus 4 ist es von Vorteil, wenn die Graphenzerlegung mit dem

Knoten s beginnt, da so frühere Zustände gelöscht werden können. Mit wenigen Anpassungen bei den s-Markierungen können aber auch Graphenzerlegungen verwendet werden, die einen anderen Startknoten besitzen.

3.4.9 Azyklische Graphen

In diesem Abschnitt wird die s,t-Zuverlässigkeit in gerichteten, azyklischen Graphen betrachtet. In azyklischen Graphen ist die Berechnung der s,t-Zuverlässigkeit einfacher als in beliebigen gerichteten Graphen. Dies ist für das Splitting des Graphen von Bedeutung. Zunächst eine Definition, was unter einem azyklischen Graphen zu verstehen ist.

Definition 3.28 Ein gerichteter Graph $G = (V, E)$ heißt azyklisch, falls er keine gerichteten Kreise enthält.

In azyklischen Graphen gilt, wie in beliebigen gerichteten Graphen, folgender Satz:

Satz 3.29 Das s,t-Zuverlässigkeitsproblem $Conn_2$ in gerichteten, azyklischen Graphen ist $\#P$ -vollständig.

Beweis: Folgt aus dem Beweis von Satz 2.10, in dem jede Kante durch einen zum Knoten t hin gerichteten Bogen ersetzt wird. Der Graph, der entsteht, ist ein gerichteter, azyklischer Graph. Die Anzahl der minimalen s,t-Schnitte in diesem Graphen ändert sich nicht. \square

Aus diesem Satz folgt unmittelbar, dass auch das s,T-Zuverlässigkeitsproblem $Conn_{sT}$ $\#P$ -vollständig ist. Für das Zusammenhangsproblem $Conn_A$ gilt dies allerdings nicht:

Satz 3.30 Das Zusammenhangsproblem $Conn_A$ in gerichteten, azyklischen Graphen ist in polynomialer Zeit lösbar.

Nach dieser Komplexitätsbetrachtung wird eine Splitting-Formel für azyklische Graphen hergeleitet. Diese Formel kann auf allgemeinere Graphen angewendet werden, falls die trennende Knotenmenge bestimmte Eigenschaften aufweist. Mit Hilfe dieser Formel kann der Ausgangsgraph in kleinere Graphen zerlegt und in diesen das Zuverlässigkeitsproblem gelöst werden. Dies führt in vielen Fällen zu einer erheblichen Reduktion der benötigten Rechenzeit. Dazu wird zunächst der Begriff der trennenden Knotenmenge eingeführt.

Definition 3.31 Sei $s, t \in V$. Eine Knotenmenge $X \subseteq V$ heißt s,t-trennend, wenn X einen Knoten von jedem s,t-Weg enthält. Dabei heißt X , mit $s, t \notin X$, eine s,t-trennende Knotenmenge (oder einfach trennende Knotenmenge), wenn es in $G - X$ keinen s,t-Weg gibt. Eine trennende Knotenmenge heißt minimal, falls sie keine trennende Kno-

tenmenge echt enthält.

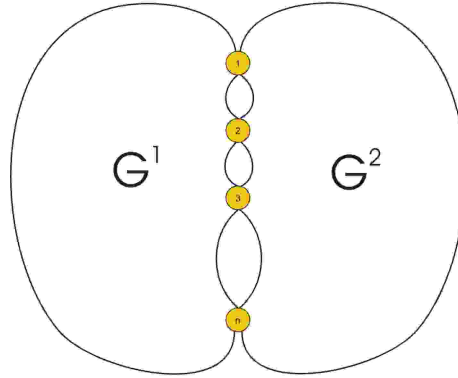


Abbildung 3.25: Trennende Knotenmenge in einem Graphen

Im Folgenden sei eine trennende Knotenmenge immer minimal. Sei nun X eine solche trennende Knotenmenge des Graphen $G = (V, E)$. Die trennende Knotenmenge teilt den Graphen G in zwei Untergraphen $G^1 = (V^1, E^1)$ und $G^2 = (V^2, E^2)$. Dabei gilt: $V^1 \cup V^2 = V$, $V^1 \cap V^2 = X$, $E^1 \cup E^2 = E$ und $E^1 \cap E^2 = \emptyset$. Nun sei der Außengrad der Knoten in X im Untergraphen G^1 , oder der Innengrad im Graphen G^2 , gleich null. Dies ist in allen azyklischen Graphen erfüllt, stellt aber eine allgemeine Bedingung an den zu untersuchenden Graphen dar. Diese Bedingung stellt sicher, dass die trennende Knotenmenge X nur einmal passiert werden kann. Sei nun $P(G^1, s, Y)$ die Wahrscheinlichkeit, dass alle Knoten aus Y von s aus erreicht werden können, aber kein Knoten in X/Y erreicht wird. Für eine Teilmenge $Y \subset X$, sei $G^2 \setminus Y$ der Graph der aus G^2 entsteht, wenn alle Knoten aus Y verschmolzen werden. Dieser neu entstandene Knoten wird im Folgenden mit y bezeichnet. Somit gibt $R(G^2 \setminus Y, y, t)$ die Wahrscheinlichkeit an, dass ein intakter Weg von mindestens einem Knoten der Menge Y zu t existiert. Die beiden soeben hergeleiteten Wahrscheinlichkeiten führen zu der folgenden Formel:

$$R_{st}(G) = \sum_{\emptyset \neq Y \subseteq X} P(G^1, s, Y) R(G^2 \setminus Y, y, t) \quad (3.24)$$

Diese Splittingformel enthält $2^{|X|} - 1$ Terme, die Schwierigkeit besteht allerdings in der Berechnung der Terme $P(G^1, s, Y)$. Der nachfolgende Satz überführt sie in einfacher zu berechnende Erreichbarkeiten.

Satz 3.32 Sei $G = (V, E)$ ein gerichteter Graph mit der trennenden Knotenmenge X . Seien $G^1 = (V^1, E^1)$ und $G^2 = (V^2, E^2)$ die zwei Trennungskomponenten. Dabei gilt, dass alle Knoten aus X in G^1 den Außengrad 0 haben. Mit $s \in V^1$ und $t \in V^2$ gilt:

$$R_{st}(G) = \sum_{Y \subseteq X} \sum_{Z \subseteq X: Z \supseteq X \setminus Y} (-1)^{|Y|+|Z|-|X|+1} R(G^1 \setminus Z, s, Z) R(G^2 \setminus Y, y, t) \quad (3.25)$$

Mit $R(G^1 \setminus \emptyset, s, \emptyset) = R(G^2 \setminus \emptyset, \emptyset, t) = 0$

Beweis: Zunächst wird gezeigt, dass in $G^1 \setminus Y$ ein Weg von s nach Y existiert, falls im Graphen G^1 zu mindestens einem Knoten von Y ein Weg existiert. Hieraus folgt für jede Teilmenge $Y \subseteq X$:

$$R(G^1 \setminus Y, s, Y) = \sum_{\substack{Y \subseteq X \\ Y \cup Z \neq \emptyset}} R(G^1, s, Z) = 1 - \sum_{Z \subseteq X \setminus Y} P(G^1, s, Z) \quad (3.26)$$

Sei nun $f(Y) = 1 - R(G^1 \setminus (X \setminus Y), s, (X \setminus Y))$. Zusammen mit der Gleichung (3.26) ergibt sich:

$$f(Y) = \sum_{Z \subseteq Y} P(G^1, s, Z)$$

Mit Hilfe der Möbiusinversion kann diese Gleichung folgendermaßen umgeformt werden:

$$\begin{aligned} P(G^1, s, Y) &= \sum_{Z \subseteq Y} (-1)^{|Y|-|Z|} f(Z) \\ &= \sum_{Z \subseteq Y} (-1)^{|Y|-|Z|} (1 - R(G^1 \setminus (X \setminus Z), s, (X \setminus Z))) \\ &= \sum_{Z \subseteq Y} (-1)^{|Y|-|Z|+1} R(G^1 \setminus (X \setminus Z), s, (X \setminus Z)) \\ &= \sum_{X \setminus Z \subseteq Y} (-1)^{|Y|+|Z|-|X|+1} R(G^1 \setminus Z, s, Z) \\ &= \sum_{Z \supseteq X \setminus Y} (-1)^{|Y|+|Z|-|X|+1} R(G^1 \setminus Z, s, Z) \end{aligned}$$

Die gewonnene Gleichung wird nun in Gleichung (3.24) eingesetzt und liefert die gesuchte Formel. \square

Beispiel 3.33 Die Anwendung der Formel (3.25) soll anhand des Graphen in Abbildung 3.26 erläutert werden. Alle Kanten haben die Ausfallwahrscheinlichkeit $q_e = 0.2$. Die

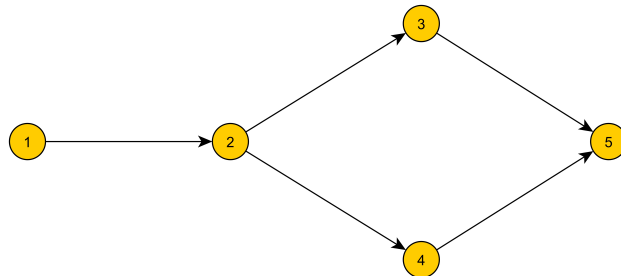


Abbildung 3.26: Beispielgraph

trennende Knotenmenge X besteht in diesem Graphen aus den Knoten 3 und 4. Somit

kann die Formel (3.25) wie folgt geschrieben werden:

$$R_{st}(G) = \sum_{Y \subseteq \{3,4\}} \sum_{Z \subseteq \{3,4\} : Z \supseteq \{3,4\} \setminus Y} (-1)^{|Y|+|Z|-1} R(G^1 \setminus Z, s, Z) R(G^2 \setminus Y, y, t)$$

Die Summe besitzt in diesem Fall sieben Summanden:

$$\begin{aligned} X = \{3\} : \quad Y = \{4\} : \quad & (-1)^1 R(G^1 \setminus \{4\}, 1, \{4\}) R(G^2 \setminus \{3\}, \{3\}, 5) \\ & Y = \{3,4\} : \quad (-1)^2 R(G^1 \setminus \{3,4\}, 1, \{3,4\}) R(G^2 \setminus \{3\}, \{3\}, 5) \\ X = \{4\} : \quad Y = \{3\} : \quad & (-1)^1 R(G^1 \setminus \{3\}, 1, \{3\}) R(G^2 \setminus \{4\}, \{4\}, 5) \\ & Y = \{3,4\} : \quad (-1)^2 R(G^1 \setminus \{3,4\}, 1, \{3,4\}) R(G^2 \setminus \{4\}, \{4\}, 5) \\ X = \{3,4\} : \quad Y = \{3\} : \quad & (-1)^2 R(G^1 \setminus \{3\}, 1, \{3\}) R(G^2 \setminus \{3,4\}, \{3,4\}, 5) \\ & Y = \{4\} : \quad (-1)^2 R(G^1 \setminus \{4\}, 1, \{4\}) R(G^2 \setminus \{3,4\}, \{3,4\}, 5) \\ & Y = \{3,4\} : \quad (-1)^3 R(G^1 \setminus \{3,4\}, 1, \{3,4\}) R(G^2 \setminus \{3,4\}, \{3,4\}, 5) \end{aligned}$$

Durch Berechnen der einzelnen Wahrscheinlichkeiten und Einsetzen in die Gleichungen, ergibt sich $R_{1,5}(G) = 0,69632$.

4 s,T-Zuverlässigkeit

Dieses Kapitel bietet einen Überblick über die Berechnung der s,T-Zuverlässigkeit in gerichteten Graphen. Diese ist eine Verallgemeinerung der, im letzten Kapitel betrachteten, s,t-Zuverlässigkeit. Bei der s,T-Zuverlässigkeit ist die Wahrscheinlichkeit gesucht, mit der es von Knoten s aus intakte Wege zu allen Knoten aus T gibt. Es folgt ein Abriss über verschiedene Reduktionsmöglichkeiten und Lösungsverfahren.

4.1 Reduktionen

Im Vorfeld der Berechnung der s,T-Zuverlässigkeit lässt sich der Graph mit Hilfe von Reduktionen verkleinern. Hier ist aber darauf zu achten, dass keine Terminalknoten entfernt werden. Dies schränkt, je nach Graph, zum Beispiel die Anwendung der Serienreduktion ein. In Abbildung 4.1 ist ein Ausschnitt aus einem Graphen zu sehen. Ohne Terminalknoten könnte dieser mit Hilfe der Serienreduktion reduziert werden. Dies ist hier aber nicht möglich, da sonst der Terminalknoten verloren gehen würde. Ist dage-



Abbildung 4.1: Konstellation bei der keine Serienreduktion möglich ist

gen der Knoten v kein Terminalknoten, kann die Serienreduktion ausgeführt werden. In Abbildung 4.2 sind die möglichen Standardreduktionen aufgeführt. Knoten, die Terminalknoten sind, sind fett gezeichnet, gestrichelte Knoten können Terminalknoten sein, müssen aber nicht.

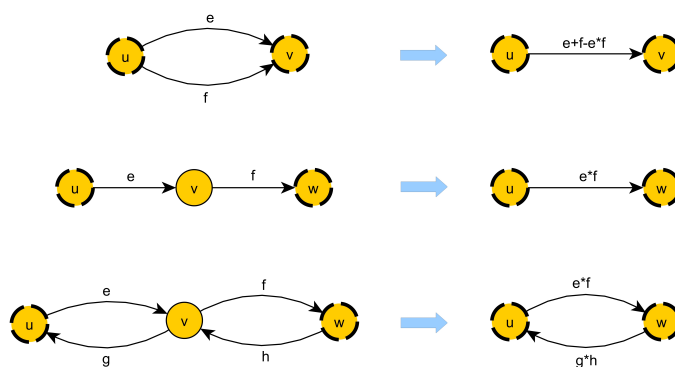


Abbildung 4.2: Reduktionen bei der Berechnung der s,T-Zuverlässigkeit

Wie bei der s,t-Zuverlässigkeit gibt es weitere, nicht zuverlässigkeitserhaltende Reduktionsmöglichkeiten (siehe auch [Kon99]). Sie sind in Abbildung 4.3 zu sehen. Für die Berechnung der s,T-Zuverlässigkeit mit Hilfe dieser drei Reduktionen gilt folgender Satz:

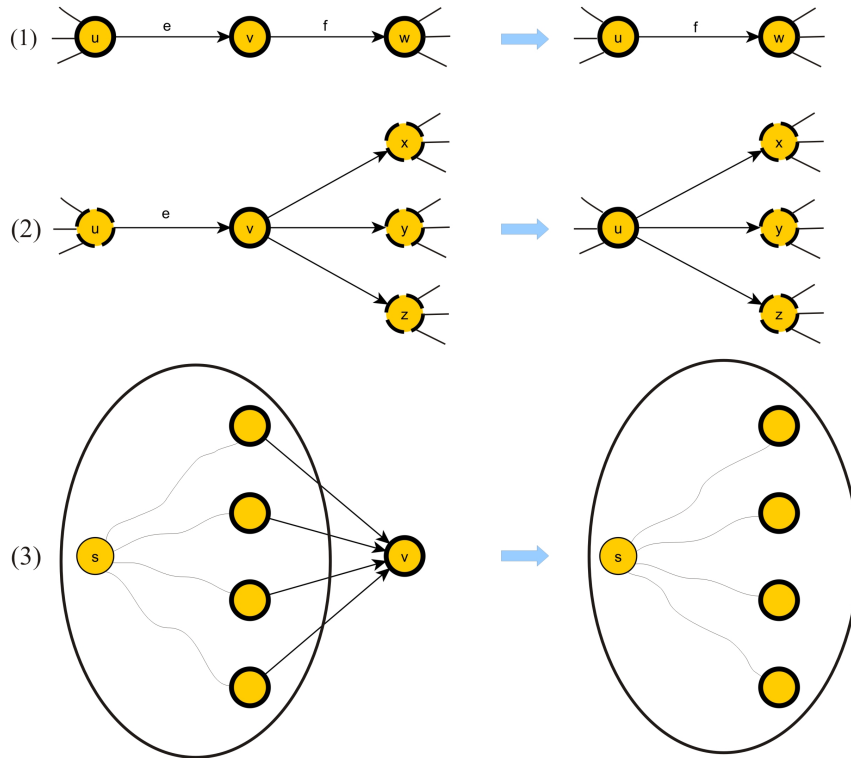


Abbildung 4.3: Reduktionen bei der Berechnung der s,T-Zuverlässigkeit

Satz 4.1 Sei $G = (V, E)$ ein gerichteter stochastischer Graph. Sei $T \subset V$ eine Menge von Knoten und $s \in V \setminus T$ ein Knoten des Graphen. Sei G' der Graph, der durch Anwendung einer der Reduktionen in Abbildung 4.3 entsteht, dann gilt für die s,T-Zuverlässigkeit:

- (1): Sei $u, v, w \in T$, dann gilt: $R_{sT}(G) = \Omega R_{sT}(G')$, mit $\Omega = p_e$ und $K' = K - \{v\}$
- (2): Sei $v \in T$, dann gilt: $R_{sT}(G) = \Omega R_{sT}(G')$, mit $\Omega = p_e$ und $K' = (K - \{v\}) \cup \{u\}$
- (3): Sei $v \in T$ und für die Menge der Nachbarn $N^-(v)$ gilt: $N^-(v) \subset T$, dann gilt: $R_{sT}(G) = \Omega R_{sT}(G')$, mit $\Omega = 1 - \prod_{e \in E^-(v)} q_e$ und $K' = K - \{v\}$. Wobei $E^-(v)$ die Menge aller Bögen ist, die in v hineingehen.

4.2 Berechnungsverfahren

In diesem Abschnitt werden einige ausgewählte Berechnungsverfahren für die s,T-Zuverlässigkeit in gerichteten Graphen vorgestellt.

4.2.1 Vollständige Enumeration

Die Herleitung und die Formeln zur vollständigen Enumeration können nahezu unverändert aus dem Abschnitt 3.3.1 zur Berechnung der s,t-Zuverlässigkeit übernommen

werden. Der Statusvektor eines Graphen $\delta = (\delta_1, \delta_2, \dots, \delta_m)$ und die Wahrscheinlichkeit eines Zustandes $P(\delta)$ sind analog zu Abschnitt 3.3.1 definiert. Um eine Aussage über die Erreichbarkeit der Knoten der Terminalknotenmenge K von s aus zu treffen, wird noch die 0,1-Variable $\phi_{s,t}(\delta)$ benötigt. Sie ist genau dann 1, wenn im Untergraphen der intakten Bögen, also der Bögen mit $\delta_i = 1$, Wege von s zu allen Terminalknoten existieren. Mit Hilfe dieser Variablen kann nun wieder die Formel zur Berechnung der s,T-Zuverlässigkeit aufgestellt werden:

$$R_{st}(G) = \sum_{\delta \in \mathcal{D}} \phi_{s,t}(\delta) P(\delta) \quad (4.1)$$

Beispiel 4.2 Für den in Abbildung 4.4 zu sehenden Graphen soll die 1,T-Zuverlässigkeit mit $T = \{3, 4\}$ berechnet werden. Alle Bögen sind mit der Zuverlässigkeit $p_e = p = 0.8$ gewichtet.

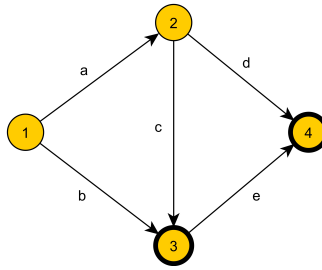


Abbildung 4.4: Graph mit zwei Nichtterminalknoten und zwei Terminalknoten

Bei diesem Graphen ist gut zu erkennen, dass das System funktionsfähig ist, falls höchstens ein Bogen ausfällt oder zwei Bögen ausfallen, bis auf die Bogenkombinationen $\{a, b\}$, $\{a, e\}$, $\{b, c\}$ oder $\{d, e\}$. Falls drei oder mehr Bögen, bis auf die Bögen $\{a, c, d\}$ ausfallen, fällt das ganze System aus. Mit diesen Informationen kann nun das Zuverlässigkeitspolynom aufgeschrieben und anschließend $q = 1 - p$ im Polynom ersetzt werden.

$$\begin{aligned}
 R_{st}(G) &= p^5 + 5p^4q + 6p^3q^2 + p^2q^3 \\
 &= p^5 - 4p^4 + 3p^3 + p^2 \\
 &= 0.86528
 \end{aligned}$$

4.2.2 Inklusion-Exklusion

Analog zur s,t-Zuverlässigkeit werden in der Formel der vollständigen Enumeration nur die Terme benötigt, bei denen $\phi_{s,t}(\delta) = 1$ gilt. Dies ist genau dann der Fall, wenn es einen gerichteten Baum im induzierten Untergraphen gibt, bei dem der Knoten s die Wurzel ist und alle Terminalknoten auf einem Ast des Baumes liegen. Im Folgenden wird ein Baum der diese Bedingung erfüllt s,T-Baum genannt. Wie im Abschnitt über die

s,t-Zuverlässigkeit kann man sich auch hier auf die Zustände beschränken, die diese Bedingung erfüllen. Es sei $\{S_1, S_2, \dots, S_k\}$ die Menge aller s,T-Bäume in G und E_i sei das Ereignis, dass alle Bögen in einem solchen Baum S_i intakt sind. Die s,T-Zuverlässigkeit ist somit die Wahrscheinlichkeit der Vereinigung der Ereignisse E_i . Es gilt somit:

$$R_{sT} = P(E_1 \cup E_2 \cup \dots \cup E_k) \quad (4.2)$$

Mit Hilfe des Inklusion-Exklusion-Prinzips kann die Gleichung (4.2) wie folgt geschrieben werden:

$$R_{sT}(G) = \sum_i P(E_i) - \sum_{i < j} P(E_i E_j) + \sum_{i < j < l} P(E_i E_j E_l) - \dots + (-1)^{k+1} P(E_1 \dots E_k) \quad (4.3)$$

Die Anzahl der Summanden in dieser Formel wächst exponentiell mit der Anzahl der dem Graphen zugrunde liegenden s,T-Bäume. Sei k die Anzahl der s-Bäume, dann besitzt die Summe 2^{k-1} Terme.

4.2.3 Dekomposition

Die aus dem Abschnitt 3.3.3 bekannte Dekompositionsgleichung kann zur Berechnung der s,T-Zuverlässigkeit verwendet werden.

$$R_{sT}(G) = (1 - p_e)R_{sT}(G - e) + p_e R_{sT}(G/e) \quad (4.4)$$

Zur Reduzierung der Dekompositionsschritte wird hier in jedem Iterationsschritt überprüft, ob von s aus zu allen Terminalknoten ein gerichteter Weg existiert. Existiert für mindestens einen Terminalknoten kein solcher Weg, muss dieser Graph nicht weiter untersucht werden. Hier können somit viel mehr Zustände ausgeschlossen werden. Bei der Dekomposition dürfen wieder nur Bögen verwendet werden, die aus der Senke s herausgehen. Wird ein Bogen kontrahiert, dessen Endknoten v in der Menge T ist, wird der Endknoten für diesen Graphen aus T entfernt, $T' = T - \{v\}$ und die s,T'-Zuverlässigkeit in diesem Teilgraphen berechnet. Vor der Anwendung des Dekompositionsverfahrens sollten wieder unnötige Knoten und Bögen gelöscht und alle möglichen Reduktionen (siehe 4.1) ausgeführt werden.

Beispiel 4.3 Die Anwendung der Dekomposition soll am Graphen in Abbildung 4.4 demonstriert werden. Der so entstehende Dekompositionsbaum ist in Abbildung 4.5 zu sehen.

Aus dem Dekompositionsbaum kann folgendes Zuverlässigkeitspolynom abgelesen wer-

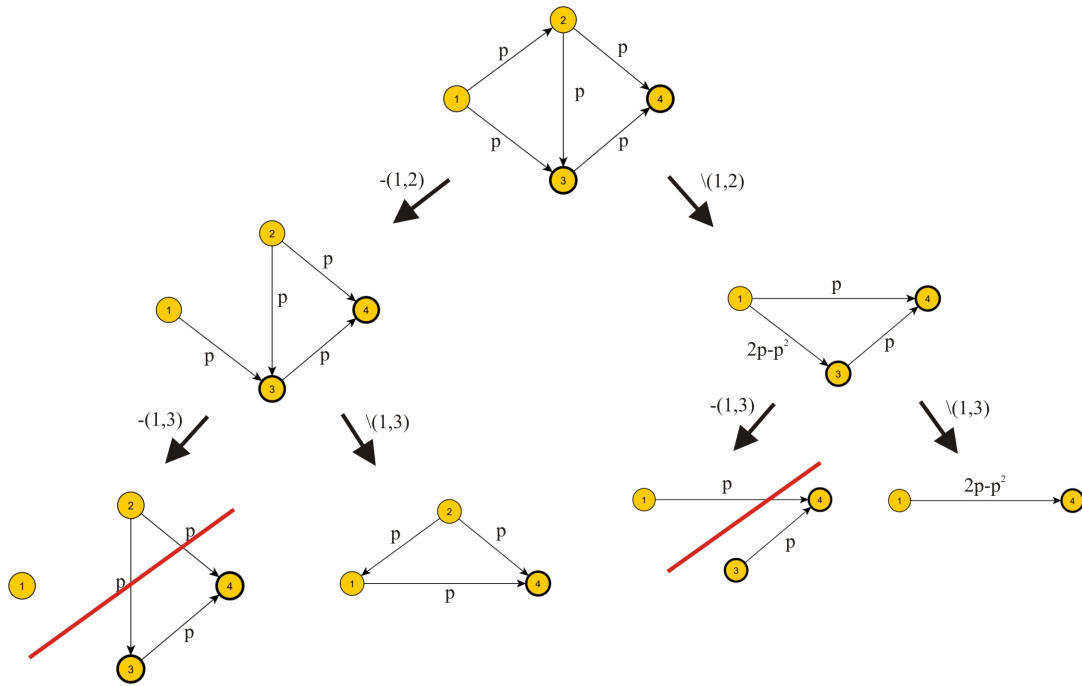


Abbildung 4.5: Dekompositionsbaum

den:

$$\begin{aligned}
 R_{sT} &= (1-p)p^2 + p(2p-p^2)(2p-p^2) \\
 &= p^5 - 4p^4 + 3p^3 + p^2
 \end{aligned}$$

4.2.4 Graphen mit beschränkter Wegweite

Zur Berechnung der s,T-Zuverlässigkeit kann eine Graphenzerlegung verwendet werden. Dieses hier vorgestellte Verfahren ist eine Erweiterung des in Abschnitt 3.4.8 vorgestellten Verfahrens zur Berechnung der s,t-Zuverlässigkeit. Es sind aber einige Erweiterungen nötig, um auch die s,T-Zuverlässigkeit berechnen zu können. Für jeden Knoten aus der Menge T wird dazu eine eigene Markierung eingeführt. Außerdem wird für jeden Zustand in einer Liste vermerkt, welcher der Knoten aus T schon abgearbeitet ist. Dabei ist ein Knoten $t_i \in T$ in einem Zustand abgearbeitet, wenn in diesem Zustand ein Knoten vorkommt, der sowohl s- als auch t_i -markiert ist, das heißt es existiert in diesem Zustand ein intakter, gerichteter Weg von s nach t_i . Diese Notierung ist notwendig, da sonst intakte Zustände nach dem Deaktivieren eines Knotens aus T gelöscht werden. Im letzten Iterationsschritt werden die Wahrscheinlichkeiten von Zuständen aufsummiert, deren einziger Knoten sowohl s-markiert ist, als auch alle T-Markierungen von Knoten besitzt, die noch nicht abgearbeitet sind. Die drei Operationen werden wie folgt ausgeführt:

1. **Knoten v aktivieren:** Im Zustand Z wird zum Graphen $G(Z)$ ein neuer Knoten v

hinzugefügt.

2. **Knoten v deaktivieren:** Im Zustand Z wird der Knoten v aus dem Graphen $G(Z)$ gelöscht. Befindet sich nach dem Löschen des Knotens v in $G(Z)$ kein s-markierter Knoten mehr, wird der ganze Zustand gelöscht, da dieser und alle nachfolgenden Zustände keinen Beitrag mehr zur s,T-Zuverlässigkeit leisten. Ein Zustand darf aber erst gelöscht werden, wenn der Knoten s schon aktiviert wurde, das heißt es wurden schon Knoten s-markiert. Falls der Knoten $t_i \in T$ schon aktiviert wurde und keine t_i -markierten Knoten mehr im Zustand existieren, kann auch hier der ganze Zustand gelöscht werden. Eine Ausnahme bildet ein Zustand, der einen Knoten v besitzt, der sowohl eine s-Markierung als auch eine t_i -Markierung besitzt. In diesem Fall wird über diesen Knoten v von s aus der Knoten t_i erreicht und somit wird der Zustand nicht gelöscht.
3. **Bogen verarbeiten:** Aus dem Zustand Z werden zwei Zustände Z' und Z'' erzeugt. Im ersten Zustand wird der Bogen nicht eingefügt und die Wahrscheinlichkeit mit $P(Z') = (1 - p_e)P(Z)$ berechnet. Im zweiten Zustand wird der Bogen (v, w) eingefügt und die Wahrscheinlichkeit mit $P(Z'') = p_e P(Z)$ berechnet. Sind beide Endknoten des Bogens gleichartig markiert, so wird kein Bogen eingefügt, da sich die Markierungen durch das Einfügen beziehungsweise das Nichteinfügen des Bogens nicht ändern. In diesem Fall ist der neue Zustand gleich dem alten Zustand, also $Z'' = Z' = Z$. Nach dem Einfügen des Bogens muss im Zustand überprüft werden, ob durch das Einfügen des Bogens Knoten markiert werden müssen.

Dieses Verfahren ist effizient zur Berechnung der s,T-Zuverlässigkeit in gerichteten Graphen mit geringer Wegweite ausführbar. In diesen Graphen ist die Berechnungsgeschwindigkeit wesentlich höher als die Berechnung mit der Dekompositionsmethode.

Beispiel 4.4 Die Anwendung der Graphenzerlegung zur Berechnung der s,T-Zuverlässigkeit soll am Graphen aus Abbildung 4.4 demonstriert werden. Eine Graphenzerlegung für diesen Graphen ist durch

$$(1, 2, (1, 2), 3, (1, 3), -1, (2, 3), 4, (2, 4), -2, (3, 4), -3, -4)$$

gegeben. Diese Graphenzerlegung führt auf den, der Tabelle 4.1 zu entnehmenden, Ablauf des Verfahrens.

Im Iterationsschritt 11 ist die Besonderheit bei der Anwendung des Graphenzerlegungsalgorithmus zu sehen. In den ersten drei Zuständen ist nach der Deaktivierung des Knotens 3 zwar keine 3-Markierung mehr vorhanden, aber der Knoten 3 wurde bereits s-markiert, somit ist er abgearbeitet. Im Zustand 4 ist nach der Deaktivierung kein Knoten mehr s-markiert. Der Zustand kann somit gelöscht werden. In den Zuständen 5 und 6 sind nach dem Deaktivieren keine Knoten mehr 3-markiert. Da der Knoten 3 noch nicht s-markiert war, können die beiden Zustände gelöscht werden.

n		G(Z)	s-markiert	T-markiert	P(Z)
0		(\emptyset, \emptyset)	\emptyset	$[\emptyset, \emptyset]$	1.0
1	+1	$([1], \emptyset)$	$\{1\}$	$[\emptyset, \emptyset]$	1.0
2	+2	$([1, 2], \emptyset)$	$\{1\}$	$[\emptyset, \emptyset]$	1.0
3	(1, 2)	$([1, 2], \{(1, 2)\})$ $([1, 2], \emptyset)$	$\{1, 2\}$ $\{1\}$	$[\emptyset, \emptyset]$ $[\emptyset, \emptyset]$	0.8 0.2
4	+3	$([1, 2, 3], \{(1, 2)\})$ $([1, 2, 3], \emptyset)$	$\{1, 2\}$ $\{1\}$	$[\{3\}, \emptyset]$ $[\{3\}, \emptyset]$	0.8 0.2
...
9	(2, 4)	$([2, 3, 4], \{(2, 3), (2, 4)\})$ $([2, 3, 4], \{(2, 3)\})$ $([2, 3, 4], \{(2, 4)\})$ $([2, 3, 4], \emptyset)$ $([2, 3, 4], \{(2, 4)\})$ $([2, 3, 4], \emptyset)$ $([2, 3, 4], \{(2, 3), (2, 4)\})$ $([2, 3, 4], \{(2, 3)\})$ $([2, 3, 4], \{(2, 4)\})$ $([2, 3, 4], \{\emptyset\})$	$\{2, 3, 4\}$ $\{2, 3\}$ $\{2, 3, 4\}$ $\{2, 3\}$ $\{2, 4\}$ $\{2\}$ $\{3\}$ $\{3\}$ $\{3\}$ $\{3\}$	$[\{2, 3\}, \{2, 4\}]$ $[\{2, 3\}, \{4\}]$ $[\{3\}, \{2, 4\}]$ $[\{3\}, \{4\}]$ $[\{3\}, \{2, 4\}]$ $[\{3\}, \{4\}]$ $[\{2, 3\}, \{2, 4\}]$ $[\{2, 3\}, \{4\}]$ $[\{3\}, \{2, 4\}]$ $[\{3\}, \{4\}]$	0.512 0.128 0.1024 0.0256 0.0256 0.0064 0.1024 0.0256 0.0256 0.0064
10	-2	$([3, 4], \emptyset)$ $([3, 4], \emptyset)$ $([3, 4], \emptyset)$	$\{3, 4\}$ $\{3\}$ $\{4\}$	$[\{3\}, \{4\}]$ $[\{3\}, \{4\}]$ $[\{3\}, \{4\}]$	0.6144 0.3136 0.0256
11	(3, 4)	$([3, 4], \{(3, 4)\})$ $([3, 4], \emptyset)$ $([3, 4], \{(3, 4)\})$ $([3, 4], \emptyset)$ $([3, 4], \{(3, 4)\})$ $([3, 4], \emptyset)$	$\{3, 4\}$ $\{3, 4\}$ $\{3, 4\}$ $\{3\}$ $\{4\}$ $\{4\}$	$[\{3\}, \{3, 4\}]$ $[\{3\}, \{4\}]$ $[\{3\}, \{3, 4\}]$ $[\{3\}, \{4\}]$ $[\{3\}, \{3, 4\}]$ $[\{3\}, \{4\}]$	0.49152 0.12288 0.25088 0.06272 0.02048 0.00512
12	-3	$([4], \emptyset)$	$\{4\}$	$[\{\}, \{4\}]$	0.86528

Tabelle 4.1: Ablauf des Graphenzerlegungsalgorithmus zur Berechnung der s,T-Zuverlässigkeit

4.2.5 Graphen mit beschränkter Baumweite

Das im vorherigen Abschnitt angegebene Verfahren ist nur auf Graphen mit geringer Wegweite effizient ausführbar. Ein Graph kann aber eine beliebig große Wegweite besitzen. Da in einem Graphen die Baumweite immer kleiner gleich der Wegweite ist, ist es sinnvoll ein Berechnungsverfahren, welches auf einer Baumzerlegung beruht, zu betrachten. Dieses Verfahren ist eine Verallgemeinerung des im letzten Abschnitt vorgestellten Verfahrens, beziehungsweise des in [Bei12] vorgestellten Berechnungsverfahrens für ungerichtete Graphen.

Grundlage des hier vorgestellten Verfahrens bildet dabei eine *schöne Baumzerlegung* des Graphen. Eine Baumzerlegung heißt *schön*, falls jeder Knoten im Baum T , mit ausgezeichneter Wurzel, maximal zwei Kinder besitzt und falls v der Sohn von u in T

ist, gilt: $\phi(u) \subseteq \phi(v)$ oder $\phi(v) \subseteq \phi(u)$ und $\phi(u)$ und $\phi(v)$ unterscheiden sich in genau einem Element. In Abbildung 4.6 ist ein Graph und eine zugehörige Baumzerlegung zu sehen.

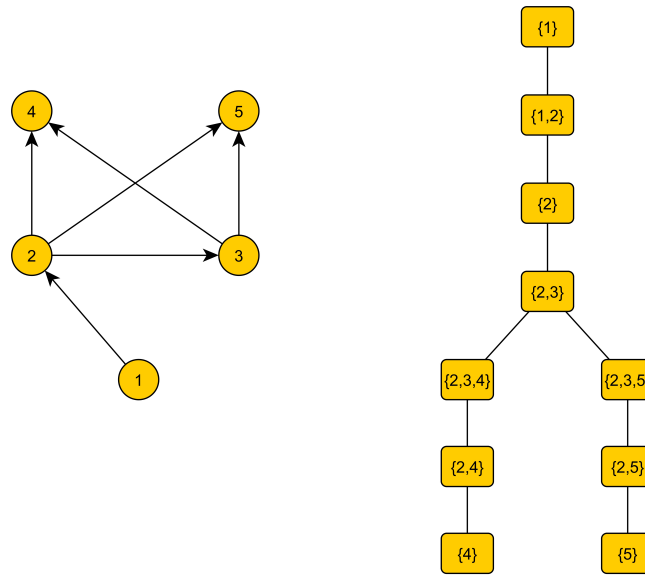


Abbildung 4.6: Baumzerlegung eines Graphen

Um aus einer solchen Baumzerlegung die s,T-Zuverlässigkeit zu berechnen, werden alle Knoten des Baumes, beginnend mit einem Blatt, verarbeitet. Da sich aufeinanderfolgende Knoten in genau einem Element unterscheiden, treten wieder die drei bekannten Operationen auf: Knoten aktivieren, Knoten deaktivieren und Bogen verarbeiten. Das Aktivieren und Deaktivieren von Knoten läuft analog zum Graphenzerlegungsalgorithmus für die s,T-Zuverlässigkeit ab. Wenn ein Knoten aktiviert wurde, werden alle nun möglichen Bögen verarbeitet. Dabei muss beachtet werden, dass jeder Bogen im Laufe der Berechnung nur einmal verarbeitet werden darf. Die Bögen werden dazu mit einer Markierung versehen, die angibt, ob der Bogen schon verarbeitet ist oder nicht. Wird ein Knoten im Baum erreicht, der zwei Kinder hat, muss zunächst die Berechnung des noch nicht betrachteten Kindes vorgenommen werden und anschließend werden die Ergebnisse der beiden Kinder verbunden. Beide Kinder bestehen aus einer Menge von Zuständen. Jeder Zustand des ersten Kindes wird nun mit jedem Zustand des zweiten Kindes verbunden. Sei $G^1 = (V^1, E^1)$ der Graph eines Zustandes des ersten Kindes und $G^2 = (V^2, E^2)$ der Graph eines Zustandes des zweiten Kindes. Es gilt: $V^1 = V^2 = V$. Für die Verknüpfung gilt: $G^{neu} = (V, E^1 \cup E^2)$. Die Markierung der Knoten des neuen Graphen wird zunächst entsprechend der ursprünglichen Markierungen vorgenommen. Nach der Bildung des neuen Graphen wird überprüft, ob noch mehr Knoten markiert werden können. Dies tritt auf, wenn in einem der beiden Ausgangsgraphen ein Bogen war, den es im anderen nicht gab. Dieser Vorgang ist in Abbildung 4.7 verdeutlicht. Die Markierungen der Knoten sind dabei anhand farbiger Striche unterhalb der Knoten dargestellt. Die Wahrscheinlichkeit des neuen Zustands berechnet sich als Produkt der Wahrscheinlichkeiten der beiden verknüpften Zustände.

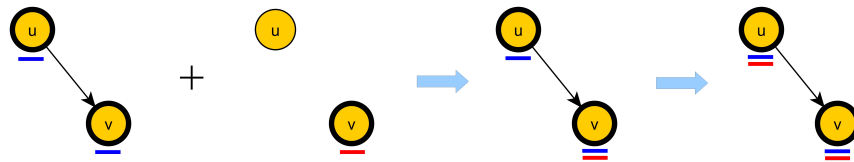


Abbildung 4.7: Verknüpfung zweier Graphen

Die Markierungen im Graphen in Abbildung 4.7 sind beides Markierungen von Knoten aus T . Zunächst werden die Markierungen aus den ursprünglichen Graphen übernommen und anschließend der Knoten u rot markiert, da vom Knoten u aus ein Knoten mit einer roten Markierung erreicht werden kann. Im letzten Iterationsschritt werden im Wurzelknoten die Wahrscheinlichkeiten aller Zustände addiert, bei denen der letzte Knoten s-markiert ist und alle noch nicht abgearbeiteten T-Markierungen besitzt.

Beispiel 4.5 Der Ablauf der Berechnung soll am Beispiel des Graphen aus Abbildung 4.6 gezeigt werden. In diesem Graphen soll die 1,4,5-Zuverlässigkeit berechnet werden. Der Ablauf ist in der Tabelle 4.2 dargestellt.

Die Berechnung startet am Knoten $\{4\}$ und folgt dem Ast des Baums in Richtung Wurzel. Am Knoten $\{2,3\}$ tritt das erste Mal ein Knoten auf, der zwei Kinder hat. Hier wird nun zunächst der zweite Ast, beginnend mit dem Knoten $\{5\}$, verarbeitet. Anschließend werden die Ergebnisse der beiden Äste, also die Zustände im Iterationsschritt 7 und 14, miteinander verrechnet. Die weitere Berechnung läuft analog zur Berechnung mittels Graphenzerlegung ab.

n		G(Z)	s-markiert	T-markiert	P(Z)
0		(\emptyset, \emptyset)	\emptyset	$[\emptyset, \emptyset]$	1.0
1	+4	$([4], \emptyset)$	\emptyset	$[\{4\}, \emptyset]$	1.0
2	+2	$([2, 4], \emptyset)$	\emptyset	$[\{4\}, \emptyset]$	1.0
3	(2, 4)	$([2, 4], \{(2, 4)\})$	\emptyset	$[\{2, 4\}, \emptyset]$	0.8
		$([2, 4], \emptyset)$	\emptyset	$[\{4\}, \emptyset]$	0.2
4	+3	$([2, 3, 4], \{(2, 4)\})$	\emptyset	$[\{2, 4\}, \emptyset]$	0.8
		$([2, 3, 4], \emptyset)$	\emptyset	$[\{4\}, \emptyset]$	0.2
...
7	-4	$([2, 3], \{(2, 3)\})$	\emptyset	$[\{2, 3\}, \emptyset]$	0.768
		$([2, 3], \{(2, 3)\})$	\emptyset	$[\{2\}, \emptyset]$	0.16
		$([2, 3], \{\})$	\emptyset	$[\{3\}, \emptyset]$	0.032
8	+5	$([5], \emptyset)$	\emptyset	$[\emptyset, \{5\}]$	1.0
...
14	-5	$([2, 3], \{(2, 3)\})$	\emptyset	$[\emptyset, \{2, 3\}]$	0.64
		$([2, 3], \{\})$	\emptyset	$[\emptyset, \{2\}]$	0.16
		$([2, 3], \{\})$	\emptyset	$[\emptyset, \{3\}]$	0.16
15	merge 7 und 14	$([2, 3], \{(2, 3)\})$	\emptyset	$[\{2, 3\}, \{2, 3\}]$	0.6144
		$([2, 3], \{(2, 3)\})$	\emptyset	$[\{2, 3\}, \{2\}]$	0.12288
		$([2, 3], \{(2, 3)\})$	\emptyset	$[\{3\}, \{2, 3\}]$	0.1024
		$([2, 3], \{(2, 3)\})$	\emptyset	$[\{2\}, \{2\}]$	0.0256
		$([2, 3], \{(2, 3)\})$	\emptyset	$[\{2\}, \{3\}]$	0.0256
		$([2, 3], \{\})$	\emptyset	$[\{3\}, \{2, 3\}]$	0.02048
		$([2, 3], \{\})$	\emptyset	$[\{3\}, \{2\}]$	0.00512
		$([2, 3], \{\})$	\emptyset	$[\{3\}, \{3\}]$	0.00512
...

Tabelle 4.2: Ablauf des Baumzerlegungsalgorithmus zur Berechnung der s,T-Zuverlässigkeit

5 Implementation und Testrechnungen

Im ersten Abschnitt dieses Kapitels wird auf die Implementation eines Frameworks zur Berechnung der s,t-Zuverlässigkeit, das im Rahmen dieser Arbeit entstanden ist, eingegangen. Es kann mit den beschriebenen Verfahren für beliebige Graphen die Zuverlässigkeit berechnen. Im darauf folgenden Abschnitt werden die Ergebnisse der Testrechnungen vorgestellt und ausgewertet.

5.1 Implementation

Zu Testzwecken und zum Vergleich der in Kapitel 3 beschriebenen Verfahren wurden diese in Python 3.2 implementiert. Das gesamte Framework *MDNetworkReliability* enthält 2226 Zeilen Code. Durch die im Python-Programm enthaltene xml-Schnittstelle zum Speichern und Laden von Graphen ist es möglich, beliebige Graphen zu laden und zu verarbeiten. Es ist weiterhin möglich, von verwendeten Graphen eine graphische Darstellung zu erzeugen. Dies geschieht mit Hilfe des Python-Paketes *pygraphviz*. *MDNetworkReliability* besteht aus drei Modulen: *MDDirectedGraph.py*, *MDNetworkReliability.py* und *tools.py*. In *MDDirectedGraph.py* ist eine Klasse für einen gerichteten Graphen implementiert. Der Graph wird hier mit Hilfe einer Adjazenzliste gespeichert. In dieser Klasse befinden sich alle nötigen Funktionen zum Testen von verschiedenen Grapheneigenschaften, wie zum Beispiel, ob der Graph azyklisch ist oder eine Artikulation besitzt. Zusätzlich enthält dieses Modul Funktionen, die nicht unmittelbar mit der Zuverlässigkeitsberechnung zu tun haben, wie die Erzeugung von Schnitten, das Erzeugen von Graphen spezieller Graphenklassen und die graphische Ausgabe von Graphen. *MDNetworkReliability.py* ist das Hauptmodul zur Berechnung der s,t-Zuverlässigkeit. Hier werden die einzelnen Unterschritte, die zur Berechnung nötig sind, aufgerufen. In diesem Modul sind auch alle in dieser Arbeit vorgestellten Verfahren zur Zuverlässigkeitsberechnung implementiert. Genauer die folgenden Verfahren: Die vollständige Enumeration (3.3.1), die Berechnung mittels Inklusion-Exklusion (3.3.2), mittels Dekomposition (3.3.3), mit Hilfe von Schnitten (3.3.4) und ein Verfahren, das eine Graphenzerlegung des Graphen benutzt (3.4.8). In *tools.py* werden verschiedene Hilfsfunktionen, wie die Vereinigung von Mengen, bereitgestellt, die in den verschiedenen Algorithmen verwendet werden.

5.1.1 Ablauf einer Berechnung

In *MDNetworkReliability.py* müssen vor dem Starten der Berechnung verschiedene Einstellungen, wie das zu verwendende Verfahren, vorgenommen werden. Hier wird auch der Graph für die Berechnung festgelegt. Er kann entweder direkt aus einer xml-Datei

importiert oder mittels Adjazenzliste angegeben werden. Durch Starten des Moduls *MD-NetworkReliability.py* wird die Berechnung der s,t-Zuverlässigkeit eingeleitet.

Die Berechnung der s,t-Zuverlässigkeit läuft im Programm in mehreren Schritten ab.

1. Laden des Graphen
2. Preprocessing
3. Starten des Berechnungsalgorithmus
4. Ausgabe des Ergebnisses und Erstellen der Zeitstatistik

Im ersten Schritt wird der Graph aus einer xml-Datei geladen oder ein manuell eingegebener Graph verarbeitet. Im zweiten Schritt, dem Preprocessing, werden im Graphen zunächst alle möglichen Reduktionen durchgeführt. Anschließend wird überprüft, ob der Graph eine Artikulation oder allgemein eine trennende Knotenmenge besitzt. Die ermittelten Informationen werden im nächsten Schritt für das Starten des entsprechenden Algorithmus verwendet. Im letzten Schritt werden die gesammelten Zeitstempel ausgewertet und eine Zeitstatistik, sowie das Zuverlässigkeitspolynom, beziehungsweise die s,t-Zuverlässigkeit, ausgegeben.

5.2 Testrechnungen

Für die Testrechnungen wurden verschiedene aus der Literatur bekannte und auch neue Graphen verwendet. In der Tabelle 5.1 ist eine Übersicht über die verwendeten Graphen angegeben. In Anhang A sind graphische Darstellungen dieser Graphen zu finden.

Name des Graphen	Anzahl der Knoten	Anzahl der Bögen	Wegweite
Bridgenetwork	4	5	2
Petersengraph	10	15	4
Prosemgraph	7	10	3
Satyanarayana-Prabhakar	9	19	4
Satyanarayana	7	12	3
PC-graph	11	22	2
FM-graph	16	29	5
Gittergraph G_8	16	22	2
Gittergraph G_{16}	32	46	2
Gittergraph G_{22}	44	130	2
Vollständiger Graph K_5	5	20	4

Tabelle 5.1: Liste der verwendeten Testgraphen

Die Testrechnungen wurden alle auf einem Rechner mit einem AMD Athlon II X4 mit 2,90GHz und 8GB Arbeitsspeicher durchgeführt. Es wurde jeweils die s,t-Zuverlässigkeit in den Graphen mit vorgegebenen Bogenzuverlässigkeiten berechnet. Für den

Startknoten s wurde immer der Knoten 1 gewählt und für den Endknoten t der Knoten mit der höchsten Ziffer. In der Tabelle 5.2 sind die Ergebnisse der Testrechnung zu sehen. Die angegebenen Zeiten geben die Gesamtzeit, also die Zeit für das Preprocessing und die Berechnung, an. Der Anteil des Preprocessings an der Gesamtzeit beträgt jedoch nur ungefähr 1%. Zellen mit einem “-” geben an, dass die Testrechnung wegen zu langer Rechenzeit abgebrochen wurden.

Name des Graphen	Vollst. Enumeration	Dekomposition	Inklusion-Exklusion	Schnittverfahren	Graphenzerlegung
Bridgenetwork	0.001	0.001	0.001	0.001	0.001
Petersengraph	0.063	0.001	0.001	0.001	0.110
Prosemgraph	0.016	0.001	0.001	0.001	0.001
Satyanarayana-Prabhakar	-	0.031	-	2.628	2.172
Satyanarayana	0.031	0.001	0.001	0.001	0.015
PC-graph	297.666	0.001	0.015	0.016	0.032
FM-graph	368.339	0.063	49.566	21.799	38.843
Gittergraph G_8	18.524	0.031	0.032	0.125	0.016
Gittergraph G_{16}	-	7.524	25.735	9.686	0.047
Gittergraph G_{22}	-	711.250	-	-	0.094
Vollständiger Graph K_5	212.614	0.001	8.098	0.609	-

Tabelle 5.2: Laufzeiten der Testverfahren in s

Die Ergebnisse der Testrechnungen zeigen, dass das Dekompositionsverfahren im Vergleich meist die besten Ergebnisse liefert. Besonders bei größeren Graphen macht sich der Laufzeitunterschied zum Schnittverfahren und dem Inklusion-Exklusion-Verfahren bemerkbar. Die Laufzeit des Schnittverfahrens hängt stark von der Struktur des Graphen ab. Im Gittergraphen G_{16} hat es nahezu die gleiche Laufzeit wie das Dekompositionsverfahren. Beim FM-Graphen beträgt die Laufzeit des Schnittverfahrens jedoch das 350-fache. In dieser Laufzeit ist auch die Zeit für das Ermitteln der Schnitte enthalten. Das Schnittverfahren ist somit nur für Graphen geeignet, die auf Grund ihrer Struktur nur wenige Schnitte besitzen. Die Bestimmung der s,t -Zuverlässigkeit mit Hilfe der Graphenzerlegung hat auf vielen Graphen zu guten Ergebnissen geführt. Hier hängt der Aufwand direkt von der Wegweite des Graphen ab. Dies ist besonders am Gittergraphen G_{22} zu sehen. Hier ist die Laufzeit des Dekompositionsverfahrens wesentlich höher. Der Gittergraph G_{22} hat zwar viele Knoten und Bögen, besitzt aber nur eine geringe Wegweite und ist somit schnell mit dem Graphenzerlegungsverfahren berechenbar. Bei den Gittergraphen fällt auch auf, dass eine Verdopplung der Größe eines Graphen auch nahezu zu einer Verdopplung der Rechenzeit führt. Bei den anderen Verfahren ist der Anstieg deutlich größer. Das Graphenzerlegungsverfahren versagt allerdings bei

Graphen mit großer Wegweite, wie am Beispiel des vollständigen Graphen K_5 oder des FM-Graphen zu sehen ist.

Die Testrechnungen haben gezeigt, dass das Dekompositionsverfahren für allgemeine Graphen eine gute Möglichkeit darstellt, die s,t-Zuverlässigkeit zu berechnen. Damit das Dekompositionsverfahren möglichst effizient ist, müssen die in den vorherigen Kapiteln vorgestellten Reduktionen in jedem Iterationsschritt angewendet werden. Auch die Verwendung von trennenden Knotenmengen führt zu einer Verbesserung der Berechnungsgeschwindigkeit. Ist bekannt, dass der Graph eine geringe Wegweite (≤ 4) besitzt, ist die Berechnung mit Hilfe der Graphenzerlegung die beste Variante. Sie hat in den Testrechnungen zu sehr guten Ergebnissen geführt. Die Graphenzerlegung ist auch bei Graphen mit sehr vielen Knoten und Bögen noch anwendbar, was bei den anderen implementierten Verfahren nicht der Fall ist. Sie eignet sich somit auch für sehr große Graphen die eine geringe Wegweite besitzen. Die vollständige Enumeration, das Inklusion-Exklusion-Verfahren und das Schnittverfahren eignen sich nicht für die Berechnung in größeren Graphen.

6 Auswertung, Schlussfolgerungen und offene Probleme

In dieser Arbeit konnten die Graphen- und Baumzerlegungsalgorithmen für die Berechnung der Zuverlässigkeit in ungerichteten Graphen auf gerichtete Graphen verallgemeinert werden. Die Testrechnungen haben bestätigt, dass mit Hilfe dieser Verfahren auch im gerichteten Fall bei Graphen mit geringer Wegweite effizient Lösungen berechnet werden können. Dabei konnten teils erhebliche Reduzierungen bei der Rechenzeit erreicht werden. Diese Algorithmen sind auch für große Netzwerke gut geeignet. Die einzige Einschränkung ist die geringe Wegweite der Graphen, die für die effiziente Anwendung der Verfahren notwendig ist. Für einige besondere Graphenklassen ist es außerdem gelungen, explizite Lösungsformeln herzuleiten. Diese expliziten Formeln können während des Berechnungsprozesses, zum Beispiel beim Dekompositionsverfahren, in Teilgraphen angewendet werden. Ferner sind in dieser Arbeit eine ganze Reihe von Reduktionsmöglichkeiten für Graphen zu finden, die besonders in der praktischen Umsetzung der Berechnungsverfahren eine große Rolle spielen. Darüber hinaus wurde gezeigt, wie das s,t -Zuverlässigkeitsproblem in der speziellen Graphenklasse der BSP-Graphen in linearer Zeit gelöst werden kann. Diese Arbeit gibt somit vielfältige Lösungsansätze zur Berechnung der s,t - und s,T -Zuverlässigkeit in gerichteten Netzwerken. Durch die parallel entstandene Implementation der Algorithmen konnte eine effiziente Umsetzbarkeit der Verfahren gezeigt werden. Die so entstandene Graphenklasse und die Berechnungsroutinen können für weitere Arbeiten und Forschungen in diesem Themengebiet genutzt werden.

Für die Effizienz der verwendeten Graphen- und Baumzerlegungsalgorithmen sind gute Weg- beziehungsweise Baumzerlegungen der Graphen von Nöten. Auch eine schnelle Überprüfung der Wegweite eines Graphen zur automatisierten Entscheidung, welches Verfahren verwendet werden soll, wäre wünschenswert. Hier besteht noch Forschungsbedarf für schnelle und einfach zu implementierende Algorithmen. Im Zusammenhang mit Zuverlässigkeitsmaßen auf gerichteten Graphen bestehen noch eine Reihe weiterer Fragen, die noch näher untersucht werden sollten. Dazu gehört insbesondere die Untersuchung von weiteren Reduktionsmöglichkeiten der Graphen. Besonders bei der s,T -Zuverlässigkeit besteht noch Forschungsbedarf. Auch nicht zuverlässigkeitserhaltende Reduktionen für die Berechnung der s,t -Zuverlässigkeit bieten noch viele Möglichkeiten der Aufwandsreduzierung. Ein weiterer Themenkomplex sind die approximativen Berechnungen der Zuverlässigkeitsmaße. Hier wäre eine Übertragung der Verfahren auf den gerichteten Fall wünschenswert. Das in dieser Arbeit vorgestellte Konzept der Berechnung der s,t - und s,T -Zuverlässigkeit mittels Baum- und Graphenzerlegung sollte auf die anderen Zuverlässigkeitsprobleme in gerichteten Graphen angewendet werden, die in dieser Arbeit nicht untersucht wurden. Ansatzpunkte für eine weitere Erforschung

sind:

- Die Berechnung von Zuverlässigkeitsmaßen mit Hilfe von PDAG (engl. Propositional Directed Acyclic Graphs)
- Die Anwendung von Monte-Carlo-Methoden zur Berechnung der Zuverlässigkeit
- Die Herleitung weiterer expliziter Formeln für spezielle Graphen
- Die Untersuchung weiterer spezieller Graphenklassen

Anhang A: Beispielgraphen

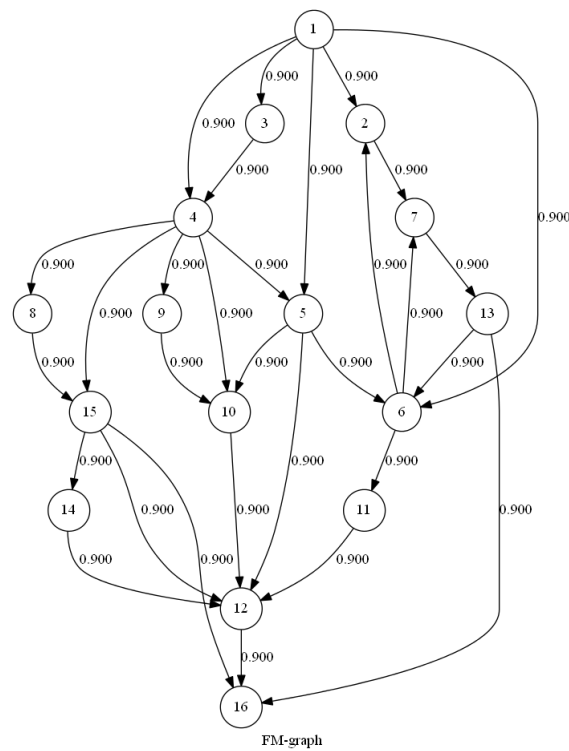


Abbildung A.1: FM-Graph

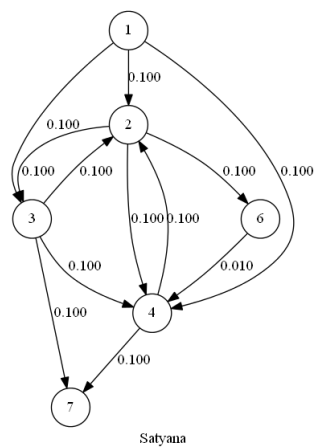


Abbildung A.2: Satyanarayana-Graph

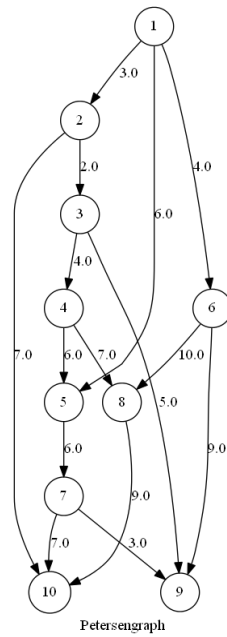


Abbildung A.3: Petersen-Graph

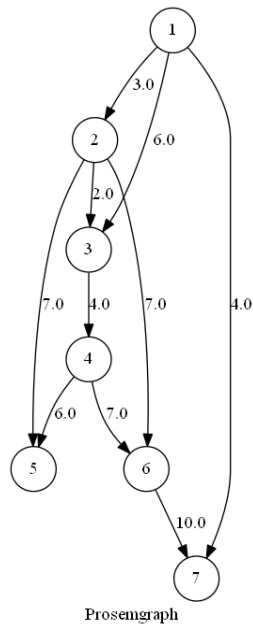


Abbildung A.4: Prosemgraph

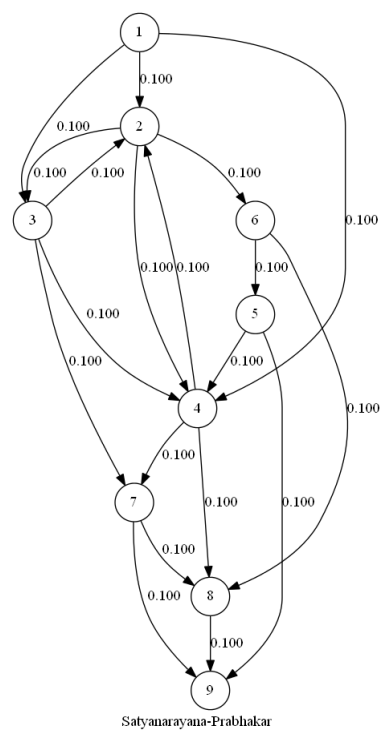


Abbildung A.5: Satyanarayana-Prabhakar-Graph

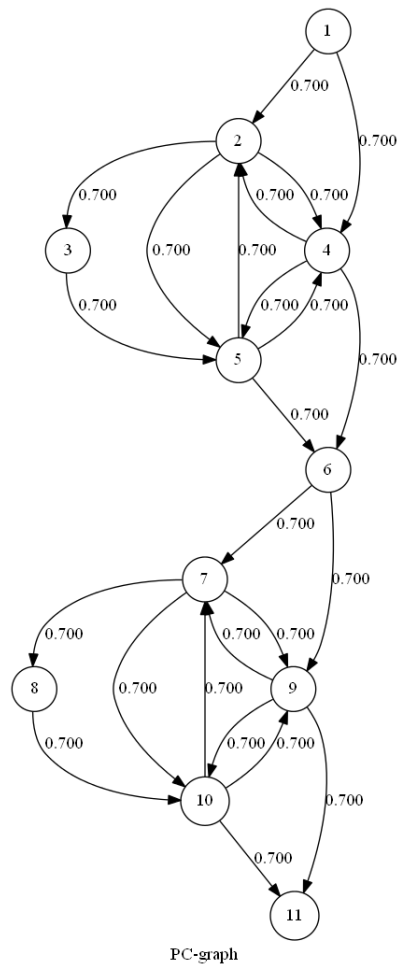


Abbildung A.6: PC-Graph

Literaturverzeichnis

- [Abo89] H.M. AboElFotouh and C.J. Colbourn. Computing 2-terminal reliability for radio-broadcast networks. *Reliability, IEEE Transactions on*, 38(5):538–555, dec 1989.
- [Abo90] H.M. AboElFotouh and C.J. Colbourn. Efficient algorithms for computing the reliability of permutation and interval graphs. *Networks*, 20(7):883–898, 1990.
- [Abr79] J.A. Abraham. An improved algorithm for network reliability. *Reliability, IEEE Transactions on*, R-28(1):58–61, 1979.
- [Agr84a] A. Agrawal and A. Satyanarayana. An $O(|E|)$ time algorithm for computing the reliability of a class of directed networks. *Operations Research*, 1984.
- [Agr84b] A. Agrawal and R.E. Barlow. A survey of network reliability and domination theory. *Operations Research*, pages 478–492, 1984.
- [Aru79] S. Arunkumar and S.H. Lee. Enumeration of all minimal cut-sets for a node pair in a graph. *Reliability, IEEE Transactions on*, R-28(1):51–55, april 1979.
- [AS85] A. Agrawal and A. Satyanarayana. Network reliability analysis using 2-connected digraph reductions. *Networks*, 15(2):239–256, 1985.
- [Bei91] F. Beichelt and P. Tittmann. A generalized reduction method for the connectedness probability of stochastic networks. *Reliability, IEEE Transactions on*, 40(2):198–204, jun 1991.
- [Bei92] W.W. Bein, J. Kambrowski, and M.F.M. Stallmann. Optimal reductions of two-terminal directed acyclic graphs. *SIAM J. Comput.*, 21(6):1112–1129, 1992.
- [Bei12] F. Beichelt and P. Tittmann. *Mathematical Models in Reliability and Maintenance*. 2012.
- [BJ08] J. Bang-Jensen and G.Z. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer Publishing Company, Incorporated, 2nd edition, 2008.
- [Bod91] H. Bodlaender and T. Kloks. Better algorithms for the pathwidth and treewidth of graphs. In Javier Albert, Burkhard Monien, and Mario Artalejo, edi-

- tors, *Automata, Languages and Programming*, volume 510 of *Lecture Notes in Computer Science*, pages 544–555. Springer Berlin / Heidelberg, 1991.
- [Bod98] H.L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, December 1998.
- [BP83] M.O. Ball and J.S. Provan. Calculating bounds on reachability and connectedness in stochastic networks. *Networks*, 13(2):253–278, 1983.
- [Bro05] J. I. Brown and Xiaohu Li. The strongly connected reliability of complete digraphs. *Netw.*, 45(3):165–168, May 2005.
- [Bro09] J. I. Brown and K. Dilcher. On the roots of strongly connected reliability polynomials. *Netw.*, 54(2):108–116, September 2009.
- [Cat96] K. Cattell, M.J. Dinneen, and M.R. Fellows. A simple linear-time algorithm for finding path-decompositions of small width. *Information processing letters*, 57(4):197–203, 1996.
- [Col87] C.J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, Inc., New York, NY, USA, 1987.
- [Cur02] N.D. Curet, J. DeVinney, and M.E. Gaston. An efficient network flow code for finding all minimum cost s-t cutsets. *Comput. Oper. Res.*, 29(3):205–219, 2002.
- [Don10] L. Dongkui. New algorithm for computing network reliability using matrix column-transformation. In *Future Networks, 2010. ICFN '10. Second International Conference on*, pages 3 –7, jan. 2010.
- [Gog04] V. Gogate and R. Dechter. A complete anytime algorithm for treewidth. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 201–208. AUAI Press, 2004.
- [Gru95] *Lehrbuch der Graphentheorie Bd. 1*. Spektrum Akademischer Verlag, 1995.
- [Hag84] J.N. Hagstrom. Using the decomposition tree for directed-network reliability computation. *Reliability, IEEE Transactions on*, R-33(5):390–395, dec. 1984.
- [Hol08] L. Holder. Graph algorithms: Applications. School of Electrical Engineering and Computer Science Washington State University, April 2008.
- [Jon07] J. Jonczyk and R. Haenni. A new approach to network reliability. *Network*, (1/23), 2007.

- [Jon08] J. Jonczy and R. Haenni. Network reliability evaluation with propositional directed acyclic graphs. *Advances in Mathematical Modeling for Reliability*, page 25, 2008.
- [Kon99] F. Kong and G. Wang. Computing the skt reliability of acyclic directed networks using factoring method. *Journal of Computer Science and Technology*, 14:56–63, 1999.
- [Kor02] M. Korenblit and V.E. Levit. The st-connectedness problem for a fibonacci graph. *WSEAS Transactions Mathematics*, 1(1):89–93, 2002.
- [Luc97] C. Lucet and J. Manouvrier. Exact methods to compute network reliability. In *Proc. of 1st International Conf. on Mathematical Methods in Reliability*. Citeseer, 1997.
- [Luc00] C. Lucet, J.F. Manouvrier, and J. Carlier. Evaluating network reliability and 2-edge-connected reliability in linear time for bounded pathwidth graphs. *Algorithmica*, 27(3):316–336, 2000.
- [Man01] E. Manzi, M. Labbe, G. Latouche, and F. Maffioli. Fishman’s sampling plan for computing network reliability. *Reliability, IEEE Transactions on*, 50(1):41–46, mar 2001.
- [Mar74] A. Martelli. *An Application of Regular Algebra to the Enumeration of Cut Sets in a Graph*. A: Pubblicazione. 1974.
- [Mar76] A. Martelli. A gaussian elimination algorithm for the enumeration of cut sets in a graph. *Journal of the ACM (JACM)*, 23(1):58–73, 1976.
- [Mil93] L.E. Miller, J.J. Kelleher, and L. Wong. Evaluation of network reliability calculation methods. 1993.
- [Moo56] E.F. Moore and C.E. Shannon. Reliable circuits using less reliable relays. I-II. *Journal of the Franklin Institute*, 262(43):191–208, 1956.
- [Mos] F. Moskowitz. The analysis of redundancy networks. In *AIEE Transactions on Communication and Electronics*.
- [Pag89] L.B. Page and J.E. Perry. Reliability of directed networks using the factoring theorem. In *Reliability, IEEE Transactions on Reliability*, volume 38, pages 556–562. 1989.
- [Poe01] A. Poenitz and P. Tittmann. Computing network reliability in graphs of restricted pathwidth, 2001.

- [Pro84] J.S. Provan and M.O. Ball. Computing network reliability in time polynomial in the number of cuts. *Operations Research*, pages 516–526, 1984.
- [Rob73] R.W. Robinson. Counting labeled acyclic digraphs. *New directions in the theory of graphs*, pages 239–273, 1973.
- [Sat78] A. Satyanarayana and A. Prabhakar. New topological formula and rapid algorithm for reliability analysis of complex networks. *Reliability, IEEE Transactions on*, R-27(2):82–100, june 1978.
- [Sat81] A. Satyanarayana and J.N. Hagstrom. Combinatorial properties of directed graphs useful in computing network reliability. *Networks*, 11(4):357–366, 1981.
- [Sat85] A. Satyanarayana and R. K. Wood. A linear-time algorithm for computing k-terminal reliability in series-parallel networks. *SIAM J. Comput.*, 14(4):818–832, 1985.
- [Sed91] R. Sedgewick. *Algorithmen*. 1991.
- [Shi86] D.R. Shier and D.E. Whited. Iterative algorithms for generating minimal cutsets in directed graphs. *Networks*, 16(2):133–147, 1986.
- [Shi91] D.R. Shier. *Network reliability and algebraic structures*. Clarendon Press, New York, NY, USA, 1991.
- [Tam11] H. Tamaki. A polynomial time algorithm for bounded directed pathwidth. In *Graph-Theoretic Concepts in Computer Science*, volume 6986 of *Lecture Notes in Computer Science*, pages 331–342. Springer Berlin / Heidelberg, 2011.
- [Tsu80] S. Tsukiyama, I. Shirakawa, H. Ozaki, and H. Ariyoshi. An algorithm to enumerate all cutsets of a graph in linear time per cutset. *J. ACM*, 27(4):619–632, October 1980.
- [Val79] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.
- [Weg03] I. Wegener. *Komplexitätstheorie: Grenzen der Effizienz von Algorithmen*. Springer Berlin Heidelberg, 2003.
- [Wol02] Thomas Wolle. A framework for network reliability problems on graphs of bounded treewidth. In *Algorithms and Computation (ISAAC 2002)*. LNCS 2518, pages 137–149. Springer-Verlag, 2002.

- [Woo85] R.K. Wood. A factoring algorithm using polygon-to-chain reductions for computing k-terminal network reliability. *Networks*, 15(2):173–190, 1985.
- [Woo89] R.K. Wood. Triconnected decomposition for computing k-terminal network reliability. *Networks*, 19(2):203–220, 1989.
- [Yan94] L. Yan, H.A. Taha, and T.L. Landers. A recursive approach for enumerating minimal cutsets in a network. *Reliability, IEEE Transactions on*, 43(3):383–388, sep 1994.
- [Yeh07] W.C. Yeh. An improved sum-of-disjoint-products technique for the symbolic network reliability analysis with known minimal paths. *Reliability Engineering & System Safety*, 92(2):260–268, 2007.
- [Yeh10] L. Yeh, B. Wang, and H. Su. Efficient algorithms for the problems of enumerating cuts by non-decreasing weights. *Algorithmica*, 56(3):297–312, 2010.
- [Yoo88] Y.B. Yoo and N. Deo. A comparison of algorithms for terminal-pair reliability. *Reliability, IEEE Transactions on*, 37(2):210–215, jun 1988.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, 20.07.2012